64'er

# Sonderheft Abenteuerspiele

SONDERHEFT 4/1986 ÖS 100,-/sfr. 14,- Lit. 12000/hfl. 18,-/dkr. 68,- DM 14,-

Markt&Technik

# 64'er

## So programmiert man Abenteuerspiele und künstliche Intelligenz

## Der Super-Kurs zum Mitmachen

- ★ mit vielen Beispiel-Listings zum Abtippen
- ★ Programmiertricks der Profis
- ★ So versteht ein Spiel mehr als 1000 Worte
- ★ Spiele ohne Speichergrenzen
- ★ So programmiert man Spiele, die denken, lernen und handeln

## Die besten Abenteuerspiele zum Abtippen

Alle Programme auch auf Diskette erhältlich

# Warum Adventure?

Wenn des Nachts jemand angespannt über einer Tastatur und einem Monitor hängt, dann ist meist ein Adventure im Spiel. Ein Adventure, ein Abenteuerspiel, ist eine Beschäftigung mit Computern, wie es keine andere gibt. Man kann mit einem Adventure in eine andere Haut schlüpfen, zum Zauberer, Weltraumpiloten, Detektiv, Barbaren oder Geheimagenten werden. Der Spieler trifft oft lebenswichtige Entscheidungen und steigert sich mit seiner Phantasie derart in die Story rein, daß er mit dem Helden zusammen die wildesten Abenteuer erlebt.

Adventures sind wie Bücher, aber ganz spezielle Bücher, denn hier kann man bestimmen, was als nächstes passiert. Adventures sind für viele wohl die schönste Art von Computerspielen.

Natürlich wollen viele Leute Adventures nicht nur spielen. Irgendwann taucht in einem der Gedankenblitz auf, selber ein Abenteuerspiel zu schreiben. Vielleicht ist es eine neue Idee, eine, die bisher in keinem Adventure verwirklicht wurde. Sie fangen an, Pläne zu machen, Skizzen anzufertigen, Grafiken auszutüfteln. Doch spätestens, wenn man sich vor den Computer setzt, kommt man ins Grübeln. Das WAS ist nicht so schwer, doch das WIE stellt einem vor Probleme. Deshalb entstand dieses Sonderheft. Michael Nickles, der einen großen Schatz an Erfahrungen in der Programmierung von Adventures besitzt, gibt sein Wissen an Sie weiter.

Michael hat es sich in diesem Heft zur Aufgabe gemacht, das Abenteuer der Adventure-Programmierung zu untersuchen, zu entschlüsseln und verständlich zu machen. Begonnen damit hat er schon vor Jahren. Die Grundbegriffe der Adventuretechnik erklärte er im 64'er-Sonderheft 2/85.

Dieses neue Heft ist allerdings keine reine Fortsetzung, denn es nutzt auch denen, die den ersten Kurs noch nicht kennen. Wer allerdings Probleme mit diesem Kurs hat, sollte vielleicht das Sonderheft 2/86 noch einmal zur Hand nehmen.

Wer nicht selber programmieren, sondern nur spielen möchte, der bekommt mit diesem Heft die besten Adventures zum Abtippen. In einem langdauernden Prozeß haben wir für Sie die interessantesten Spiele herausgesucht, mit denen Sie voll auf Ihre Kosten kommen werden. Wir hoffen, daß Sie mit dem großen Kurs und den ausgezeichneten Programmen auch auf den Geschmack gekommen sind und jetzt das Rüstzeug besitzen, um ebenfalls gleich gute oder sogar noch bessere Adventures zu schreiben. Lassen Sie es uns ruhig wissen und scheuen Sie sich nicht, sie uns zuzuschicken. Wer wagt, gewinnt.

(Georg Klinge/Boris Schneider)

# PROGRAMM-SERVICE

# 64'er

Bestellungen in der Schweiz: Markt&Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Tel. 042/41 56 56
Bestellungen in Österreich: Bücherzentrum Meidling, Schönbrunner Straße 261, A-1120 Wien, Tel. 0222/83 31 96,
Microcomput-ique E. Schiller, Fasangasse 21, A-1030 Wien, Tel. 0222/78 56 61,
Ueberreuter Media Handels- und Verlagsgesellschaft mbH, Alser Straße 24, A-1091 Wien, Tel. 0222/48 15 38-0
Bestellungen aus anderen Ländern bitte per Auslandspostanweisung!

## Das Angebot dieser Ausgabe:

Wer keine Zeit oder keine Lust hat, alle Programme selbst in mühevoller Kleinarbeit abzuschreiben, kann wieder auf den bewährten Programm-Service zurückgreifen. Alle Programme, die mit dem Diskettensymbol im Inhaltsverzeichnis gekennzeichnet sind, gibt's auf Diskette.

**Bestell-Nr. L6 86 S4D**
**2 Disketten**

**DM 34,90***
**(sFr. 29,50/öS 239,-*)**



* inkl. MwSt. Unverbindliche Preisempfehlung

## Programme aus früheren Ausgaben:

**Ausgabe 3/86**
**Bestell-Nr. L6 86 03D** Diskette
**DM 29,90* (sFr. 24,90/öS 299,-*)**

| | |
|---|---|
| Eingabehilfe Checksummer V3 und MSE | S. 55 |
| Kudiplo – Funktion diskutieren und plotten | S. 58 |
| 64'er DOS – alle Funktionen der 1541 beschleunigen | S. 65 |
| Shapes auf dem C64 mit Demo-Programm | S. 71 |
| Auto-Old: letzte Rettung nach »new« | S. 79 |
| Englisch für Fortgeschrittene | S. 80 |
| HiRes-Scrolling mit Demo-Programm und Quelltext | S. 81 |
| 1520-Plotter als Drucker | S. 82 |
| Laufschriftgenerator – ruckfreie Laufschrift für eigene Programme | S. 83 |
| Centronics-Interface mit Quelltext für den C128 | S. 84 |
| View Picture – Endlich auch farbige Hi-Eddi-Bilder für eigene Programme | S. 91 |

**Ausgabe 2/86**
**Bestell-Nr. L6 86 02D** Diskette
**DM 29,90* (sFr. 24,90/öS 299,-*)**

| | |
|---|---|
| text-transposer | S. 51 |
| Garbage Collection: Müllabfuhr für Strings in max. 1 Sekunde | S. 53 |
| Eingabehilfe: MSE + Checksummer | S. 57 |
| Profiauflösung für MPS 801/803 | S. 59 |
| Software zum 64'er Eprom-Programmiergerät | S. 65 |
| Spitzmon: Der Monitor zum Ascompiler Basic und Compilerversion | S. 69 |
| Tips und Tricks für Profis | S. 77 |
| Sound-Editor | S. 80 |
| CIA: Echtzeituhr/DFÜ | S. 98 |
| Schreiberling: Märchenstunde für Drucker MPS 801/802/803 | S. 102 |

**Ausgabe 1/86**
**Bestell-Nr. L6 86 01D** Diskette
**DM 29,90* (sFr. 24,90/öS 299,-*)**

| | |
|---|---|
| Checksummer V3 | S. 54 |
| MSE V1.0 | S. 54 |
| Datawork 1.1 | S. 56 |
| Ascompiler | S. 60 |
| Hardcopy | S. 67 |
| Life | S. 69 |
| Vergleich von Programmen | S. 77 |
| MSE-Hex-Tastatur | S. 78 |
| Die unmögliche Uhr | S. 78 |
| Screenlarger + Demo | S. 81 |
| C128 – Grafikprogramme: | |
| – Fensterrose | S. 131 |
| – Spiralen | S. 133 |
| – Box-Befehl | S. 134 |
| IEEE-Generator | S. 147 |

**Ausgabe 12/85**
**Bestell-Nr. L6 85 12D** Diskette
**DM 29,90* (sFr. 24,90/öS 299,-*)**
**Bestell-Nr. L6 85 12K** Kassette
**DM 29,90* (sFr. 24,90/öS 299,-*)**

| | |
|---|---|
| Checksummer V3 | S. 54 |
| MSE V1.0 | S. 54 |
| Old für C128 | S. 43 |
| Chemie-Assistent | S. 57 |
| SMU | S. 68 |
| Hyperscreen | S. 76 |
| Grafik-80 | S. 80 |
| Seeschlacht | S. 93 |
| Eprom-Automat | S. 93 |
| Tipp-Utility | S. 99 |
| Floppymonitor | S. 105 |
| Auto.OBJ | S. 108 |
| Bildsch.Langsam | S. 107 |
| Taschenrechner | S. 107 |
| Code-ASCII | S. 107 |
| 88-Zeichen | S. 107 |
| Frogger | S. 106 |
| Scroll n. unten | S. 106 |
| Zahlenraten | S. 108 |
| Auto-Befehl | S. 107 |
| SWAP | S. 153 |
| BSP.-Quelltext | S. 169 |

**Ausgabe 11/85**
**Bestell-Nr. L6 85 11A**
**DM 29,90* (sFr. 24,90/öS 299,-*)**

| | |
|---|---|
| Checksummer V3 | S. 54 |
| MSE | S. 54 |
| Koala-Painter Hardcopy | S. 39 |
| Lyrik-Maschine (AdM) | S. 55 |
| Hypra-Platos (LdM) | S. 61 |
| Profiprint | S. 71 |
| Apfelmännchen | S. 80 |
| Block Out | S. 84 |
| Spritekill | S. 86 |
| Screen-Dump | S. 88 |
| Pseudo-IRQ | S. 88 |
| INPUT-Routine | S. 90 |
| Synthetische Melodien | S. 95 |
| Hypra-Ass Ergänzung | S. 96 |
| Reassembler | S. 97 |
| Vier Betriebssysteme | S. 105 |
| Spiralen | S. 151 |
| HiRes-Spiralen | S. 151 |

**Ausgabe 10/85**
**Bestell-Nr. L6 85 10A**
**DM 29,90* (sFr. 24,90/öS 299,-*)**

**Ausgabe 9/85**
**Bestell-Nr. L6 85 09A**
**DM 29,90* (sFr. 24,90/öS 299,-*)**

**Ausgabe 8/85**
**Bestell-Nr. L6 85 08A**
**DM 29,90* (sFr. 24,90/öS 299,-*)**

**Ausgabe 7/85**
**Bestell-Nr. L6 85 07A**
**DM 29,90* (sFr. 24,90/öS 299,-*)**

**Ausgabe 6/85**
**Bestell-Nr. L6 85 06A**
**DM 29,90* (sFr. 24,90/öS 299,-*)**

**Ausgabe 5/85**
**Bestell-Nr. L6 85 05A**
**DM 29,90* (sFr. 24,90/öS 299,-*)**

**Ausgabe 4/85**
**Bestell-Nr. L6 85 04A**
**DM 29,90* (sFr. 24,90/öS 299,-*)**

**Ausgabe 3/85**
**Bestell-Nr. L6 85 03A**
**DM 29,90* (sFr. 24,90/öS 299,-*)**

**Ausgabe 2/85**
**Bestell-Nr. L6 85 02A**
**DM 29,90* (sFr. 24,90/öS 299,-*)**

**Ausgabe 1/85**
**Bestell-Nr. L6 85 01A**
**DM 29,90* (sFr. 24,90/öS 299,-*)**

**Sonderheft 3/86 – C16, C116, VC20, Plus 4**
1 Diskette für VC20 und C16/116:
**Bestell-Nr. L6 86 S3 CD**
**DM 29,90* (sFr. 24,90/öS 299,-*)**
1 Kassette für VC20:
**Bestell-Nr. L6 86 S3 KV**
**DM 19,90* (sFr. 17,-/öS 199,-*)**
1 Kassette für C16:
**Bestell-Nr. L6 86 S3 KC**
**DM 19,90* (sFr. 17,-/öS 199,-*)**

**Sonderheft 2/86 – Tips&Tricks**
**Bestell-Nr. L6 86 S2D** Diskette
**DM 29,90* (sFr. 24,90/öS 299,-*)**

**Sonderheft 1/86 – C128er**
**Bestell-Nr. L6 86 S1D** Diskette
**DM 29,90* (sFr. 24,90/öS 299,-*)**

**Sonderheft 8/85 – Assembler**
**Bestell-Nr. L6 85 S8D** Diskette
**DM 29,90* (sFr. 24,90/öS 299,-*)**
**Bestell-Nr. L6 85 S8K** Kassette
**DM 19,90* (sFr. 17,-/öS 199,-*)**

**Sonderheft 7/85 – Professionelle Anwendungen**
**Bestell-Nr. L6 85 S7D** 2 Disketten
**DM 34,90* (sFr. 29,50/öS 349,-*)**
**Bestell-Nr. L6 85 S7K** 4 Kassetten
**DM 34,90* (sFr. 29,50/öS 349,-*)**

**Sonderheft 6/85 – Top-Themen**
**Bestell-Nr. L6 85 S6** 2 Disketten
**DM 34,90* (sFr. 29,50/öS 349,-*)**

**Sonderheft 5/85 – Floppy, Datasette**
**Bestell-Nr. L6 85 S5D** Diskette
**DM 29,90* (sFr. 24,90/öS 299,-*)**
**Bestell-Nr. L6 85 S5K** Kassette
**DM 19,90* (sFr. 17,-/öS 199,-*)**

**Sonderheft 4/85 – Grafik**
**Bestell-Nr. L6 85 S4A**
**DM 29,90* (sFr. 24,90/öS 299,-*)**

**Sonderheft 3/85 – Spiele**
**Bestell-Nr. L6 85 S3 A** 2 Disketten
**DM 34,90* (sFr. 29,50/öS 349,-*)**

**Sonderheft 2/85 – Abenteuerspiele**
**Bestell-Nr. L6 85 S2**
**DM 34,90* (sFr. 29,50/öS 349,-*)**

**Sonderheft 1/85 – Tips&Tricks**
**(2. überarb. Auflage)**
**Bestell-Nr. CB 023** Floppy-Utilities
**DM 29,90* (sFr. 24,90/öS 299,-*)**
**Bestell-Nr. CB 024** Hilfsprogramme
**DM 29,90* (sFr. 24,90/öS 299,-*)**

* inkl. MwSt. Unverbindliche Preisempfehlung.

Bitte verwenden Sie für Ihre Bestellung und Überweisung die eingeheftete Postgiro-Zahlkarte, oder senden Sie uns einen Verrechnungs-Scheck mit Ihrer Bestellung.
Sie erleichtern uns die Auftragsabwicklung, und dafür berechnen wir Ihnen keine Versandkosten.

# INHALT

## Vorwort



## Dateiverwaltung



## Künstliche Intelligenz



## Parser-Programmierung



## Adventures grafisch aufwerten



## Adventure-Programmierung



## Eingabehilfen



## Adventurelistings



## Nachschau

# Willkommen im Land des Abenteuers

**Abenteuerspiele sind eine fantastische Sache. Nicht nur für Spieler, sondern auch für Programmierer eröffnet sich ein weites Feld für kreative Schöpfungen. Wir möchten Ihnen das notwendige Rüstzeug dazu geben.**

Lieber Adventure-Freund, auf den folgenden Seiten möchte ich Ihnen gerne erläutern, worum es in diesem Kurs geht – wo Schwerpunkte gesetzt werden, was Sie Neues lernen werden etc. Einige von Ihnen werden sich vielleicht schon mit dem Abenteuerspiele-Programmierkurs, der im Rahmen des 64'er-Sonderheftes 2/85 veröffentlicht wurde, beschäftigt haben. Wer will, kann den folgenden Kurs als eine Fortsetzung beziehungsweise Ergänzung des ersten Kurses betrachten. Lassen Sie mich noch einmal kurz zusammenfassen, worum es im ersten Programmierkurs ging:

Im ersten Abschnitt wurde erklärt, was Adventures sind, wie man sich diese Spiele ausdenkt, sich auf die Programmierung derselben vorbereitet und schließlich, wie man Spielpläne erstellt.

Im folgenden Abschnitt wurde dann ausführlich beschrieben, wie man komfortable Befehlseingabe-, Codier- und Analyseprogramme schreibt. So entstand schließlich ein Programmodul, das in der Lage war, eingegebene Befehlssätze wie zum Beispiel »Nimm das Schwert und geh nach Norden« in einzelne Worte zu zerlegen, um es anschließend in VERB-, OBJEKT-, GEGENSTANDS- und PERSONENzahl zu codieren.

Im zweiten Hauptteil des Kurses ging es dann um die Actionprogrammierung in Abenteuerspielen – also wie man Spielhandlungen programmiert.

Am Ende dieses Kurses war schließlich jeder Teilnehmer (hoffe ich doch!) in der Lage, Adventures mit den folgenden Eigenschaften zu programmieren:

- Verständnis von ganzen deutschen Sätzen
- Realtime (computergesteuerte Personen wie zum Beispiel Monster im Spiel)
- Programmieren der Adventures ausschließlich mit 38 KByte Basic-RAM
- Speichermedium: Datasette ausreichend.

Kurzum, mit dem anhand des Kurses erworbenen Wissen konnte man die Art von Adventure mit relativ gutem Sprachverständnis (Parser) programmieren.

Wozu also nun wieder ein Adventure-Programmierkurs?

Um es gleich auf einen Nenner zu bringen: Die damals erlernten Programmiertechniken entsprechen nicht mehr dem neuesten Stand derer, die heute unter Einsatz von Floppy-Laufwerken etc. möglich sind.

Während man den ersten Kurs noch komplett mit C 64 und Datasette bewältigen konnte, ist es nun unbedingt erforderlich, mit einem Floppy-Laufwerk zu arbeiten. Man beachte lediglich die Tatsache, daß heuer fast keine professionellen Adventures mehr auf Kassette veröffentlicht werden. Dies ist nicht nur deshalb der Fall, weil die Datasette zu langsam ist, sondern vielmehr darin zu begründen, daß die Datenmengen, die für ein Super-Adventure benötigt werden, nur noch mit Hilfe eines Disketten-Laufwerks zu bewältigen sind. Beachten Sie doch einmal die folgende Tatsache:

Ein C 64 mit Datasette verfügt im Endeffekt über 64 KByte RAM, die man ordentlich verwalten kann (sequentielle Dateien auf Band sind für die Programmierung von großen Adventures unbrauchbar).

## Viel Speicher für Adventures

Ein C 64 mit Floppy-Laufwerk im Gegensatz kann mit den 64 KByte RAM des C 64 und einer Disk-Seite, die eine Kapazität von zirka 170 KByte hat, einen Datenbereich von 234 KByte verwalten. Schon heute gibt es allerdings Adventures wie zum Beispiel Dragonworld, Fahrenheit 451 und Amazon, die auf 4 Diskseiten gespeichert sind und somit einen Umfang von 4 mal 170 KByte – also 680 KByte haben. Diese riesige Speicherkapazität kommt letztendlich durch die zahlreichen Grafiken zustande, die diese Adventures der Firma Telarium aufweisen. Würde man auf Grafik verzichten (à la Infocom), hätte man 680 KByte für Text zur Verfügung. 680 KByte – das sind nicht mehr und nicht weniger als 200 vollgeschriebene A4-Schreibmaschinenseiten! Eine Textfülle dieser Art ist bisher jedoch noch nie bei einem Adventure aufgetreten. Werfen wir doch einmal einen kurzen Blick auf die Infocom-Textadventures. Diese Spiele sind lediglich auf einer Disk-Seite abgespeichert, das heißt, die Spiele verfügen über maximal 60 Seiten Text (wobei ein Filecompactor schon mitberechnet ist). Wenn es nun doch schon möglich ist – warum schreibt denn niemand Super-Adventures mit unglaublichem Textumfang? Ganz einfach – Adventure-Spiele entwerfen und gestalten ist viel Arbeit. Kaum ein Programmierer würde ein Infocom-ähnliches Text-Adventure aus reinem Enthusiasmus beziehungsweise »just for fun« programmieren, da die dafür benötigte Arbeitszeit nicht zu bezahlen wäre – oder würden Sie vielleicht 500 Mark für ein Adventure ausgeben, das den 5fachen Textumfang eines 100-Mark-Programms hat und Sie auch 5mal so viel Zeit zur Lösung kosten würde?

Vielleicht ja, aber der Programmierer muß auch die Tatsache berücksichtigen, daß sein Spiel wahrscheinlich schon wenige Tage nach seinem Erscheinen geknackt wird und somit illegal auf den Markt kommt. Auch ist es sehr schwer abzuschätzen, ob ein Super-Adventure, an dem man etwa ein Jahr lang arbeitet, nach seiner Fertigstellung immer noch etwas Besonderes darstellt, beziehungsweise ob es dann noch aktuell ist. Kurz gesagt, sogenannte Super-Adventures mit riesigem Textumfang, deren Realisierung zwar schon möglich ist, wird es wohl auch in absehbarer Zeit nicht geben. Ein gutes Adventure, das heute auf den Markt kommt, sollte folgende Eigenschaften aufweisen können:

- Guter Textumfang.
- Eventuell ausgezeichnete HiRes-Grafiken.
- Ein Parser, der über ein Vokabular von über 1000 Wörter verfügt und auf jedes Spielkommando intelligent reagiert – also nie einfach nur stupide Antworten gibt wie zum Beispiel »Das geht nicht« auf ein Kommando wie etwa »Nimm das Zimmer«, sondern vielmehr auch begründet, warum dies nicht geht – also zum Beispiel »Sie befinden sich selbst im Zimmer und können es somit auch nicht nehmen«.
- Das Spiel sollte einen so großen Umfang haben, daß ein durchschnittlicher Abenteuerfan zirka 30-40 intensive Stunden zum Lösen des Spiels benötigt.

Ein neuer Trend ist es, Bücher in Adventures umzusetzen. Ein solches Adventure braucht nicht unbedingt viel knifflige Rätsel enthalten, die den Spieler in Verzweiflung bringen. Es geht im Prinzip nur darum, ein Buch in einem neuen Medium

– also dem Computer – zu präsentieren. Ein Spielen dieses Adventures kann somit gleich dem Lesen des Buches gesetzt werden – mit der Ausnahme jedoch, daß sich der Spieler frei »im Buch« bewegen kann und somit noch mehr mit in die Handlung der Geschichte einbezogen wird. Solche Spiele werden dann auch meistens unter Mitarbeit der entsprechenden Buchautoren in Abenteuerspiele umgesetzt.

Zwei erwähnenswerte Spiele dieser Art sind Infocoms »Hitchhikers Guide to The Galaxy« sowie Telariums »Fahrenheit 451«. Besonders lobenswert ist die Umsetzung von Douglas Adams »Hitchhikers Guide«. Wer die Bücher gelesen hat und dann das Spiel spielt, wird viele identische Textstellen des Buches wiederfinden. Trotzdem hilft es einem im Spiel kaum weiter, wenn man die Bücher kennt, denn hier werden witzige Aufgaben gestellt, die zwar voll und ganz dem Buchstil entsprechen, aber trotzdem nur durch viel Knobeln gelöst werden können. Ein Beispiel dafür ist zum Beispiel eine sprechende Tür in einem Raumschiff. Ich stehe vor der Tür und gebe als Kommando »open door« ein. Die Tür jedoch antwortet ganz frech, daß sie nur demjenigen Durchlaß gewährt, der seine Intelligenz beweisen könnte. Mittels »Inventory« erfahre ich schließlich, daß ich neben vielem Krimskrams und dem Bademantel, mit dem ich bekleidet bin, auch noch eine Zahnbürste bei mir habe.

Gelangweilt gebe ich den meiner Ansicht nach unsinnigen Kommentar »Show door the toothbrush« ein, ohne zu hoffen, daß dieser Kommentar vom Programm verstanden wird. Zur Verblüffung jedoch teilt die Tür mir freundlich mit: »That's only a normal toothbrush – everyone can have that!«. Mehr zu den Infocom-Spielen später.

Wenn Sie sich für die Programmierung dieser neuen Adventure-Generation interessieren, dann ist der nun folgende Kurs genau das Richtige für Sie.

## Benötigte Grundlagen

Wer diesen Kurs komplett durcharbeiten möchte, benötigt unbedingt einen C 64 oder C 128 mit Disketten-Laufwerk. Im Prinzip ist kein spezielles Grundwissen erforderlich, abgesehen von der Praxis im Umgang mit Basic natürlich.

Der Kurs enthält zwar zahlreiche Maschinenroutinen, die jedoch so gut beschrieben werden, daß jeder sie benützen und in eigenen Programmen anwenden kann. Wer diese Routinen gemäß eigenen Vorstellungen modifizieren will, muß sich allerdings mit Assembler auskennen. Leichter wird sich noch derjenige tun, der sich mit dem Speicheraufbau – also der Speicherlandschaft des C 64 auskennt. Zwar werde ich auch hier kurz auf die Grundlagen eingehen, es ist jedoch sehr empfehlenswert, vorher Kurse aus der Zeitschrift 64'er wie zum Beispiel den Grafik-Programmierkurs oder »Memory-Map mit Wandervorschlägen« durchgearbeitet zu haben. Wer sich mit Basic im Rahmen von Stringoperationen etc. noch nicht so gut zurechtfindet, dem möchte ich an dieser Stelle noch den entsprechenden Programmierkurs im ersten Abenteuerspiele-Sonderheft empfehlen.

Wie baut sich der folgende Kurs auf? Wie bereits erwähnt, werden drei große Bereiche behandelt: Dateiverwaltung, Programmierung von künstlicher Intelligenz und Adventure-Programmierung für Fortgeschrittene.

Es bietet sich daher an, einmal zu betrachten, in welchem Zusammenhang diese drei Bereiche stehen, beziehungsweise warum gerade diese drei Bereiche für diesen Kurs ausgesucht wurden.

Dateiverwaltung: Wie es der Name schon sagt, geht es hier um die Verwaltung von Dateien – also irgendwelchen Datenmengen. Effektive Dateiverwaltung ist nur mit einem großen Speicher möglich. Dies ist auch der Grund, warum hierzu unbedingt eine Diskettenstation erforderlich ist.

Was hat Dateiverwaltung denn nun mit der Programmierung mit Abenteuerspielen zu tun? Ganz einfach – betrachten wir doch einmal ein Textabenteuerspiel. Der reine Text bezieht sich auf folgende Dinge:

- Text als Raumbeschreibung
- Text als Beschreibung von Objekten
- Text als Beschreibung von Aktion.

Hierzu einige Beispiele:

| | |
|---|---|
| Raumbeschreibung: | Land of shadow<br>You are in a land of dark shadows and shallow hills, wicht stretch out in all directions. To the west, the land dips sharply. |
| Objektbeschreibung: | A strage grey machine, shaped somewhat like a clothes dryer, is on one side of the room. |
| Aktionsbeschreibung: | A familiar voice calls down to you. »Are you still there?« he bellows with a coarse laugh. »Well, then, grab onto the rope and we'll see what we can do.« The rope drops to within your reach. |



Diese Beispiele habe ich – wie es alle Adventure-Freaks sicherlich bemerkt haben – dem Infocom Adventure ZORK III entnommen.

Was ich anhand dieser Beispiele zeigen möchte, ist, daß bei Text-Adventures der gesamte Text in drei Bereichen gegliedert wird, der dann mittels Tabellen verwaltet wird. So gibt es zum Beispiel unter anderem eine Raumbeschreibungstabelle, in der genau steht, an welcher Stelle auf der Diskette (auf der der Text gespeichert ist) sich die Raumbeschreibung zu jedem im Spiel vorkommenden Raum befindet. Angenommen, der Computer benötigt die Beschreibung zum Raum X. Er schaut dann einfach in seiner Tabelle nach, wo sich auf Disk die Beschreibung zu Raum X befindet.

Nun benötigt er noch ein Unterprogramm, das in der Lage ist, den Text von Raum X anhand des von der Tabelle ermittelten Zeiger-Wertes von der Diskette zu lesen.

Ob man nun Adressen, Lagerbestandsteile oder Raumbeschreibungen auf Diskette speichern und verwalten will, spielt keine Rolle – die Grundlagen der Dateiverwaltung sind auf alle Bereiche anwendbar.

Auch im Rahmen der künstlichen Intelligenz (oder abgekürzt KI) ist Dateiverwaltung unbedingt erforderlich.

**KI-Programme:** Gerade KI-Programme benötigen einen großen Vorrat, aus dem sie ihr »Wissen« beziehungsweise ihren »Einfallsreichtum« schöpfen können. Ich beziehe mich hierbei weniger auf Rechenprogramme als Mastermind-Knacker, sondern vielmehr von ELIZA-artigen Programmen –

als Programme, die in der Lage sind, menschliche Befehle und Fragen zu interpretieren, zu analysieren, um schließlich eine logische Antwort beziehungsweise intelligente »Verlegenheitsaussage« zu geben.

**Abenteuerspiele:** Auch bei Abenteuerspielen erlangten bei der Programmierung die Dateiverwaltung und die Künstliche Intelligenz immer größer werdende Rollen. Ich spreche hierbei von den sogenannten Super-Adventures, also Adventures mit großem Textumfang und fantastischem Sprachverständnis.

Nun aber zum eigentlichen Kursaufbau.

## Kursaufbau

**Dateiverwaltung:** Hier lernen Sie alle Möglichkeiten der Dateiverwaltungsprogrammierung kennen. Es werden Dateien innerhalb von Basic behandelt, sequentielle Dateien auf Disk und Band, relative Dateien auf Disk sowie Handhabung von index-sequentiellen Dateien. Außerdem erfahren Sie Interessantes über verschiedene Sortieralgorithmen sowie Suchroutinen, die später zum Verwalten des Wortschatzes benötigt werden. Im Rahmen dieses Abschnitts wird auch ein Zeichen-Editor vorgestellt, mit dem wir für die späteren Kapitel einen deutschen Zeichensatz erstellen werden. In einem besonderen Kapitel werde ich auch auf alle Möglichkeiten des Textraffens zu sprechen kommen – also, wie man Text so kürzt, daß man möglichst viel davon auf einer Disk-Seite unterbringen kann.

Unterstützt wird dieser Kurs-Bereich durch einige kurze, aber äußerst effektvolle Maschinenprogramme, die die Programmierung von Dateiverwaltungen erheblich unterstützen.

Krönung dieses Kursabschnitts ist schließlich ein anwendungsfreundlicher Texteditor, mit dem es möglich ist, Dateien auf Disk zu erstellen (auch speziell in bezug auf Text-Adventures), Dateien auf andere Dateien zu übertragen, Dateien zu sortieren, Dateien auf Drucker auszugeben etc. Dieser Super-Texteditor wird für den ganzen Kurs ein unentbehrliches Hilfsmittel sein.

**Adventures:** Im Rahmen des Adventure-Kursteils lernen Sie alles, was man zur Programmierung von Super-Adventures wissen muß. Sie erfahren zunächst, wie man Texte für Adventures ausarbeitet und mittels dem Texteditor auf einer Diskette als relative Datei speichert.

Kern dieses Kursabschnittes ist jedoch zweifellos der Parser 5.0. Mit Hilfe dieses Parsers ist es möglich, Adventures zu schreiben, die mit über 1000 gespeicherten Wörtern nahezu jeden Spielerbefehl verstehen können – also fast den gesamten deutschen Grundwortschatz. Sie erfahren, wie der Parser 5.0 direkt mit dem Texteditor zusammenarbeitet und der Clou: Trotz des unglaublich großen (bei deutschen Text-Adventures bisher zweifellos unerreichten Vokabulars) bleiben dem Programmierer satte 38 KByte Basic-RAM zur absolut freien Verfügung!

*Das heißt, jeder kann mit Hilfe des Texteditors und den Parser 5.0 sowie einigen Maschinenroutinen mit einfachen Basic-Kenntnissen Super-Adventures programmieren, wie es sie zur Zeit noch gar nicht gibt!*

Damit aber noch nicht genug. Im Rahmen der Adventure-Programmierung wird noch ein anderes kurzes, aber äußerst effektvolles Programm präsentiert: Der Adventure-Cracker – ein Programm, mit dem die Lösung fast jeden Adventures zum reinsten Kinderspiel wird – einfach Adventure-Cracker-Programm starten, Diskette eines Adventures (zum Beispiel Dragonworld, Fahrenheit 451 etc.) einlegen und sich den gesamten Textinhalt des Adventures auf Drucker beziehungsweise Bildschirm ausgeben lassen!

Selbstverständlich lernen Sie gleich anschließend, wie Sie Ihre eigenen Adventures gegen solche Text-Cracker absolut schützen können: Mit Hilfe einer Textverschlüsselung.

Wer glaubt, daß damit genug wäre, irrt – auch die Grafik-Freaks kommen diesmal voll und ganz auf Ihre Kosten. Ich werde ausführlich erläutern, wie Grafik-Bilder der professionellen Spiele entstehen (zum Beispiel die fantastischen Grafiken der Telarium-Adventures). Neben Grundlagen erfahren Sie einiges über Bilder-Digitalisierung etc. Jeder kennt sie – verschiedene Malprogramme wie Koala-Painter und Paint Magic, die zwar alle ihre Vorteile haben, jedoch untereinander nicht kompatibel sind. Auch hier wird mit einigen Übertragungsprogrammen endgültig Abhilfe geschaffen. HiRes-Bilder der bekanntesten Malprogramme in eigenen Adventures einbauen? – kein Problem!

Und zu guter Letzt: Ein Maschinenprogramm, das auf dem Bildschirm ein Text- und ein Grafikfenster darstellt. Anhand einiger Maschinenroutinen können Sie dann zum Beispiel ein Bild mit einem Digitizer erstellen, auf Paint-Magic-Format umwandeln, neu bearbeiten, dann ins Fensterformat umwandeln, um es letztendlich ins eigene Adventure einzubauen.

**KI-Programme:** Dieser Kursbereich bezieht sich natürlich auf das Programmieren von künstlicher Intelligenz (englisch: artificial intelligence). Es werden Programme erläutert, die diverse Spiele wie etwa NIM spielen können. Dies hat eigentlich nicht mehr sehr viel mit Adventures und Dateiverwaltung zu tun, ist als Einführung zu diesem Thema jedoch nicht wegzudenken. Interessant wird es allerdings, wenn wir zusammen Programme schreiben, die menschliche Befehlseingaben verstehen, interpretieren und darauf antworten können. Wenngleich wir auch erstaunliche Resultate erzielen werden, so sollten Sie Ihre Erwartungen nicht zu hoch stecken! Auch die geschicktesten Programmiermethoden können aus Ihrem C64 keinen emotionellen Gesprächspartner für einsame Stunden oder lange Winterabende machen. Es wird auch nicht möglich sein, den C64 innerhalb dieses Kurses im Rahmen der KI total bis ins letzte Bit auszuschöpfen. Um Programme zu schreiben, die den gesamten deutschen Grundwortschatz beherrschen, müßte man zunächst einmal diesen Grundwortschatz in den Computer eingeben. Daß dieser Wortschatz jedoch nicht so einfach in dieser Zeitschrift zum Abtippen gedruckt werden kann, dürfte wohl jedem klar sein. Ich werde Ihnen zwar einige sehr nette Eliza-artige Programme vorstellen, aber wer sich an ganz große Projekte heranwagen will, muß dies mit viel Eigeninitiative tun. Selbstverständlich wird er auch dann im Kurs viele Tips und Tricks, die zum Angehen eines solchen Programms nötig sind, erfahren.

## ...und vieles andere mehr

Hiermit dürften Sie einen guten Einblick in den Aufbau und Inhalt des Kurses haben. Natürlich werden noch viele Tips und Tricks auftauchen, die ich an dieser Stelle noch nicht erwähnt habe. Wer glaubt, daß dieser Kurs nüchtern und langweilig wird, muß seinen Irrtum spätestens dann zugeben, wenn wir in Kürze folgendes Problem gelöst haben werden:

Halten Sie es für möglich, daß in manchen Fällen ein gewöhnlicher Spiegel ein unentbehrliches Hilfsmittel zum Programmieren eines Computers sein kann?
Denken Sie mal nach!

Nun genug der langen Einleiterei. Bitte führen Sie folgende Aktionen in der richtigen Reihenfolge aus:

- Kühlschrank öffnen, Kühlmittel beschaffen (Cola, Bier, etc.)
- Grundnahrungsmittel bereitstellen (Kartoffelchips)
- Computer, Floppy, Monitor einschalten
- Sessel zurechtrücken und gemütlich machen
- Auf Seite 10 blättern
- Kurs durcharbeiten

Los geht's....

(M. Nickles/bs)

## Hier mein »Lebenslauf«:

Geboren wurde ich am 18.9.1966 in Erlenbach am Main. Bis zu meinem 10. Lebensjahr besuchte ich den Kindergarten und anschließend die Grundschule. Dann begann die schwärzeste Zeit meines Lebens, der ich in zirka 3 Monaten mit dem Abitur ein Ende bereiten werde: Ich kam ins Gymnasium. Eines der wenigen positiven Dinge, die mir dort begegneten, war ein PET 2001, an dem ich mit 14 Jahren meine ersten Programmierkenntnisse sammelte. Ich war damals ein absoluter Computerfreak und verbrachte die meisten Nachmittage mit diesem PET. Mein Traumcomputer war damals der Atari 800, den ich mir aufgrund des hohen Preises (ca. 2000 Mark) nicht leisten konnte. Drei Jahre später hatte ich endlich genug Geld für einen Homecomputer gespart: natürlich den C 64. Ich stellte schnell fest, daß Computer ein teures Hobby sind, weil man mit der Zeit alle nur erdenklichen Pheripheriegeräte haben will. Um mit dem C 64 Geld zu verdienen, begann ich Adventures zu schreiben. Nach zahlreichen Wegwerf-Programmen gelang es mir, ein wirklich gutes Adventure zu schreiben: Gordon Saga I. 1984 ging ich mit diesem Programm zur Commodore-Messe in Frankfurt. Bei Happy Software fand ich schnell einen Vertrag für das Spiel. Gleichzeitig lernte ich dort einen 64'er-Redakteur kennen. Ich erzählte ihm von meinen Erfahrungen im Bereich Adventureprogrammierung, und es entstand so letztendlich mein erster Programmierkurs im 64'er-Sonderheft 2/85. Jetzt, ein Jahr später, schrieb ich meinen zweiten Kurs, den Sie im vorliegenden Sonderheft finden.

Trotz meiner aktiven Beschäftigung mit Computern bezeichne ich mich keineswegs als Computerfreak, da ich noch zahlreiche andere Hobbys habe: Ich interessiere mich fürs Filmemachen und bin aktives Mitglied der besten Punk-Gruppe (Musik) der Welt: Brutallica.

Den größten Teil meiner Freizeit verbringe ich im Freundeskreis auf Parties, in Kneipen, Konzerten, im Kino etc. Kälte kann ich nicht ausstehen und fahre bzw. trampe so oft es geht in südliche Länder wie Frankreich und Italien. Mein derzeitiges Idol ist der Programmierer vom Davids Midnight Magic – er hat angeblich schon seinen zweiten Porsche 911...

Ich wünsche Ihnen viel Spaß und Erfolg beim Lesen meines zweiten Programmierkurses, der voraussichtlich das Letzte ist, was ich mit meinem C 64 gemacht habe. Mein nächster Computer ist der Atari 520 ST.

(Michael Nickles)

# Dateiverwaltung komplett

**Wenn der Speicherplatz im Computer nicht mehr reicht, muß man auf das Diskettenlaufwerk ausweichen. Wir zeigen, wie man Daten sinnvoll und effektiv verwalten kann.**

Die einfachste Methode, Dateien innerhalb eines Basic-Programmes zu verwalten, ist, den Text (im Fall einer Datei für Texte) einfach in DATA-Zeilen beziehungsweise dimensionierten Stringfeldern abzulegen.

Zunächst wollen wir einmal von einem ganz einfachen Beispiel ausgehen: Wie auch in späteren Kapiteln dieses Kurses stellen wir uns hierzu eine Aufgabe beziehungsweise geben uns eine Datei vor, an der wir anschließend arbeiten werden.

Gehen wir einmal von folgender Datei aus:

```
01  BANANE
02  ORANGE
03  TRAUBE
04  APFEL
05  BIRNE
06  COCA COLA
07  ERDBEERE
08  MANDARINE
09
10  HIMBEERE
11  PFLAUME
12  BIRNE
13  BROMBEERE
```

Zugegeben, dies ist keine besonders glückliche Datei (nicht etwa, weil sie 13 Elemente enthält), sondern vielmehr eine ungeordnete Liste von Früchten. Einige von Ihnen wundern sich bestimmt auch schon, was Amerikas bestverkauftes Getränk in dieser Liste zu suchen hat – dazu jedoch später.

Lassen Sie mich einmal kurz aufzählen, was für eine ideale Datei charakteristisch ist:

1. Die Datei bezieht sich auf genau einen Bereich, beziehungsweise ist in einzelne Gruppen unterteilt. Das heißt, in einer Datei, mit der man zum Beispiel Videokassetten ordnen und verwalten will, haben Elemente wie Stromabrechnung oder Zeitschrifteninhaltsverzeichnisse nichts zu suchen. Genauso ist Coca Cola in unserer Beispielsdatei, mit der wir arbeiten wollen und die sich auf das Verwalten von Früchten bezieht, total fehl am Platz.
2. Eine Datei kann in einzelne Gruppen unterteilt werden. Wenn man zum Beispiel eine Datei zum Verwalten eines Haushaltes eröffnen möchte, kann man sie in die Gruppen Stromabrechnung, Wasserabrechnung, Telefongebühren etc. unterteilen. Diese einzelnen Gruppen stehen dann in festgelegten Bereichen der Haushaltsdatei und dürfen sich nicht überschneiden.
3. Es bietet sich an, die Dateien nach bestimmten Kriterien wie zum Beispiel alphabetisch zu sortieren.

Unsere Beispielsdatei sieht auf den ersten Blick zwar wie eine relativ vernünftige Aufzählung von Früchten aus, ist jedoch noch lange keine ideale Datei in bezug auf die obigen drei Definitionen.

## Aufbau einer Datei

Hier einige Betrachtungen zum Aufbau unserer Früchtedatei:
- Die Datei besteht aus 13 SÄTZEN. Unter einem SATZ verstehen wir eine Information (zum Beispiel APFEL in unserer Früchtedatei), die an einer bestimmten Stellen der Datei zu finden ist. Entprechend unserer Beispielsdatei könnte man also sagen »SATZ Nummer 1 heißt BANANE« oder »BANANE steht in SATZ 1 der Früchtedatei« beziehungsweise »BANANE steht an erster STELLE in der Früchtedatei«.
- In der Früchtedatei sind drei Beerenarten vertreten – ERDBEERE, HIMBEERE und BROMBEERE.
- Satz 9 der Datei ist leer beziehungsweise an Stelle 9 der Früchtedatei ist eine Lücke.
- In SATZ 6 steht ein Element (COCA COLA), das nicht in diese Datei gehört (da COCA COLA ein Getränk ist und in unserer Früchtedatei nur Früchte vorkommen sollen).
- Die Datei ist weder alphabetisch noch nach einem anderen erkennbaren Kriterium sortiert beziehungsweise geordnet.

Bevor wir mit unserer Datei arbeiten können, müssen wir sie zunächst einmal in den Computer bringen.

Wir legen die »Früchte« hierzu in DATA-Zeilen ab und lesen sie dann in ein dimensioniertes Stringfeld D$ ein (Listing 1).

Mit der Anweisung GOSUB 10000 wird unsere Früchtedatei definiert und kann dann bearbeitet werden.

```
10000 REM FRUECHTE-DATEI DEFFINIEREN -----
      ------------------------------------
      --                                      <026>
10010 SZ=13     :REM ANZAHL DER SAETZE       <133>
10020 DIM D$(SZ):REM STRING-FELD             <163>
10030 FOR I=1 TO SZ                           <073>
10040 : READ D$(I)                            <073>
10050 NEXT I                                  <228>
10100 DATA BANANE                             <123>
10105 DATA ORANGE                             <023>
10110 DATA TRAUBE                             <132>
10115 DATA APFEL                              <045>
10120 DATA BIRNE                              <009>
10125 DATA COCA COLA                          <060>
10130 DATA ERDBEERE                           <245>
10135 DATA MANDARINE                          <050>
10140 DATA                                    <248>
10145 DATA HIMBEERE                           <172>
10150 DATA PFLAUME                            <066>
10155 DATA BIRNE                              <044>
10160 DATA BROMBEERE                          <226>
10190 RETURN                                  <086>
```

**Listing 1. Definition der Früchtedatei**

```
10 GOSUB 10000: REM DATEI DEFFINIEREN       <103>
100 REM STEUERPROGRAMM -------------------
    ------------------------------------- <020>
110 INPUT"WELCHEN SATZ ";SA                 <054>
120 IF SA<0 OR SA>SZ THEN PRINT"DIESER SAT
    Z EXISTIERT NICHT!":GOTO 110            <098>
130 PRINT:PRINT D$(SA)                      <112>
140 GOTO 100                                <068>
```

**Listing 2. Lesen eines Datensatzes**

```
10 GOSUB 10000: REM DATEI DEFFINIEREN       <103>
100 REM STEUERPROGRAMM -------------------
    ------------------------------------- <020>
110 X$="":INPUT"WELCHEN BEGRIFF SOLL ICH S
    UCHEN ";X$                              <254>
120 FOR SA=1 TO SZ                          <039>
130 : IF X$=D$(SA) THEN PRINT"DER GESUCHTE
     BEGRIFF STEHT IN SATZ ";SA             <008>
140 NEXT SA                                 <068>
150 GOTO 110                                <094>
```

**Listing 3. Suche nach einem Datensatz**

## Erste Gehversuche

Wir wollen nun einige Experimente mit unserer Datei durchführen.

Das Unterprogramm zur Definition ist immer das gleiche. Sie müssen lediglich jeweils die anderen Programmzeilen anpassen.

**SATZ lesen**

Unser erstes »Experiment« ist so einfach, daß sich jegliche Erklärungen erübrigen. Bitte ergänzen Sie nun um die Zeilen aus Listing 2.

Dieses Steuerprogramm verlangt also die Eingabe von Satzzahlen und gibt daraufhin den entsprechenden Satz aus.

**SATZ suchen**

Hier geht es darum, einen bestimmten SATZ innerhalb der Datei zu finden. Man gibt als Suchbegriff zum Beispiel ERDBEERE ein und erhält als Antwort SATZ 7 – also der gesuchte Begriff ERDBEERE steht in SATZ 7 der Früchtedatei. Das Steuerprogramm hierfür finden Sie in Listing 3.

Auch dieses Programm ist noch sehr einfach. Nach Eingabe des gesuchten Begriffs als X$ durchläuft eine Schleife SA die gesamte Datei D$(1) bis D$(SZ) – wobei SZ der höchste SATZ der Datei ist – und überprüft bei jedem Schritt, ob X$ gleich D$(SA) ist. Wird eine Übereinstimmung gefunden, so erhält man den entsprechenden Satzwert SA.

```
10 GOSUB 10000: REM DATEI DEFFINIEREN               <103>
100 REM STEUERPROGRAMM --------------------
    ---------------------------------------- <020>
110 X$="":INPUT"WELCHEN BEGRIFF SOLL ICH S
    UCHEN ";X$                               <254>
120 IC=0:FOR SA=1 TO SZ                      <168>
130 : IF X$=D$(SA) THEN PRINT"DER GESUCHTE
     BEGRIFF STEHT IN SATZ ";SA:IC=IC+1      <101>
140 NEXT SA                                  <068>
142 IF IC=0 THEN PRINT"DIESER BEGRIFF IST
    IN DER FRUECHTEDATEI NICHT ENTHALTEN!"   <045>
144 IF IC>1 THEN PRINT"ACHTUNG, DER BEGRIF
    F ";X$;" KOMMT";IC;"MAL VOR!"            <087>
150 GOTO 110                                 <094>
```

**Listing 4. Suchen eines Datensatzes**

```
10 I$="ABCDEFGHIJKLMNOPQRSTUVWXYZ"           <098>
20 X$="":PRINT"{CLR}"                        <230>
30 X$=X$+"BEI DER BUCHSTABENFOLGE "          <044>
40 X$=X$+LEFT$(I$,INT(23*RND(1))+1)          <076>
50 X$=X$+" HANDELT ES SICH UM EINEN TEIL D
   ES ALPHABETS."                            <071>
60 PRINT X$                                  <166>
70 PRINT"{2DOWN}BITTE EINE TASTE DRUECKEN!
   "                                         <212>
80 POKE 198,0:WAIT 198,1                     <052>
90 GOTO 20                                   <020>
```

**Listing 5. Das PRINT-Problem**

```
20000 REM FORMATIERTE TEXTAUSGABE ---------
      -------------------------------------
      --                                     <251>
20010 IF X$=""THEN 20090                     <204>
20020 IF LEN(X$)<40 THEN PRINT X$:GOTO 200
      90                                     <239>
20030 IF LEN(X$)=40 THEN PRINT X$;:GOTO 20
      090                                    <125>
20040 I$=LEFT$(X$,41):IC=0:FOR I=1 TO LEN(
      I$):IF MID$(I$,I,1)=" "THEN IC=I       <247>
20050 NEXT I:IF IC=0 THEN PRINT X$:GOTO 20
      090                                    <062>
20060 I$=LEFT$(I$,IC-1):IF LEN(I$)=40 THEN
       PRINT I$;                             <051>
20070 IF LEN(I$)<40 THEN PRINT I$            <121>
20080 X$=RIGHT$(X$,LEN(X$)-IC):GOTO 20010    <243>
20090 RETURN                                 <080>
```

**Listing 6. Formatierte Textausgabe**

Im Falle einer idealen Datei würde dieses Steuerprogramm optimal funktionieren. Bei unserer momentan noch etwas chaotischen Datei kann es jedoch noch zu Schwierigkeiten kommen. Betrachten wir einmal folgende Fälle:

– Suchen Sie doch einmal den Begriff »PFIRSICH«. Selbstverständlich werden Sie keine Antwort erhalten, da diese Frucht in unserer Früchtedatei nicht aufgeführt wird. Unser Steuerprogramm kann also insofern verbessert werden, daß es mit »Dieser Begriff ist in der Früchtedatei nicht enthalten!« antwortet. Diese Änderung läßt sich noch einfach vornehmen, aber es gilt auch noch ein anderes Problem zu lösen:

– Geben wir als Suchbegriff »BIRNE« ein, so erhalten wir zwei Antworten. Einmal, daß der Begriff in SATZ 5 steht und einmal in SATZ 12. Diese Antwort des Steuerprogramms ist völlig korrekt, da der Begriff beziehungsweise die Frucht BIRNE in unserer Früchtedatei zweimal auftritt – nämlich in den SÄTZEN 5 und 12.

Allerdings ist es ziemlich unsinnig, bei unserer Früchtedatei eine Frucht zweimal aufzuführen. Warum habe ich die Frucht BIRNE eigentlich zweimal aufgeführt, sollten Sie sich jetzt fragen. Ganz einfach – wenn man sehr lange Listen eintippen muß (zum Beispiel Ersatzteile-Bestandsliste für einen Kfz-Betrieb) kann es sehr leicht passieren, daß man einen Satz versehentlich mehrmals eintippt. Spätestens das Dateiverwaltungs-Programm sollte dann soviel »Intelligenz« (Anführungszeichen deshalb, da dies noch lange nichts mit KI (künstliche Intelligenz) zu tun hat) haben, um den Fehler zu erkennen und auf ihn aufmerksam zu machen. Es gibt selbstverständlich auch Dateiverwaltungen, bei denen es sinnvoll ist, Begriffe doppelt auszuführen (zum Beispiel Objekt- und Wortschatztabellen für Abenteuerspiele) – dazu jedoch später.

Wir wollen unser derzeitiges Steuerprogramm also noch so verbessern, daß es einen Hinweis gibt, wenn ein Begriff mehrmals auftaucht.

Versuchen Sie zur Übung einmal diese beiden Verbesserungen am Steuerprogramm selbst durchzuführen, bevor Sie die Lösung abtippen. Die Lösung finden Sie in Listing 4.

Dieses Steuerprogramm ist schon relativ brauchbar. Es kann Begriffe innerhalb unserer Datei suchen, erkennt, wenn ein gesuchter Begriff in der Datei überhaupt nicht vorkommt und es teilt uns mit, wenn ein Begriff in der Datei mehrmals enthalten ist (zum Beispiel BIRNE).

Bisher haben wir bei unseren Beispielprogrammen noch keinen Wert auf eine schöne Darstellung beziehungsweise Formatierung der Textausgaben auf dem Bildschirm geachtet. Im Laufe des Kurses wollen wir uns jedoch auch diesem Problem zuwenden.

Wenn man bei unserer Früchtedatei den Begriff BIRNE sucht, so erhalten wir die Antwort »Achtung, der Begriff kommt 2mal vor!«.

Auf dem Monitor sieht dieser Satz allerdings nicht sehr gut dargestellt aus, wenn man einmal das Wort »vor« am Ende des Satzes betrachtet. Das »v« des Wörtchens steht am Ende einer Bildschirmzeile, während das »or« zu Beginn der nächsten Zeile steht.

Viel deutlicher noch wird das Problem einer formatierten Textausgabe im Beispiel von Listing 5.

Dieses Programm macht nichts anderes, als einfach einen zufällig erzeugten Text auf dem Bildschirm auszugeben. Das heißt, Anfang und Ende des Textes sind immer gleich, der mittlere Teil stellt jedoch eine zufällige, aus dem Alphabet ausgewählte Buchstabenfolge dar. Durch die zufällige Auswahl des mittleren Textteils variiert die Gesamtlänge des Textes stark – es kommt so zu einer unkontrollierten, unformatierten Bildschirmausgabe. Einige Computer haben in ihrem Basic PRINT-Anweisungen mit »Intelligent Space« zur Verfügung, mit deren Hilfe es sehr einfach ist, Text formatiert auf den Bildschirm zu bringen. Eine derartige Routine ist im Basic des

Commodore 64 leider nicht enthalten. Sie läßt sich jedoch mit nur geringem Aufwand als Basic-Unterprogramm programmieren, wie es Listing 6 zeigt.

Nun wollen wir dieses Unterprogramm in unser letztes Beispielsprogramm einbauen. Hierzu müssen lediglich folgende Änderungen durchgeführt werden:

```
60 GOSUB 20000
```

Wenn Sie das Programm nun erneut starten, werden alle Textausgaben formatiert auf den Bildschirm gebracht. Was die formatierte Textausgabe anbelangt, so wollen wir es zunächst bei diesem Unterprogramm belassen.

## Eine MERGE-Routine

Wie Sie sicherlich bereits bemerkt haben, kommen in diesem Kurs Routinen vor, auf die wir ständig zurückgreifen werden. Um sich viel Tipparbeit zu ersparen, bietet es sich daher an, die einzelnen Routinen einzeln abzuspeichern, um sie bei Bedarf wieder laden zu können. Hierbei ist es nützlich, wenn Sie mit einem MERGE-Befehl arbeiten, der es erlaubt, einzelne Programmteile nacheinander zusammen »zu laden«. Sie können hierzu Basic-Erweiterungen à la G-Basic, Simons Basic oder auch Toolkits à la KFC Super 64 verwenden. Mit fortlaufendem Kurs werden wir jedoch soviel Speicherplatz benötigen beziehungsweise das Betriebssystem des C64 derartig »umkrempeln« und erweitern, daß es sich mit keiner dieser Basic-Erweiterungen mehr verträgt. Es bietet sich sodann die folgende MERGE-Routine, die Sie im Direkt-Modus durchführen können, an (sowohl für Disklaufwerk als auch Datasette).

Dieser Direktmodus-MERGE geht so:

A. Programm A laden. Dann notieren Sie sich den Inhalt der Speicherzellen 43 und 44 – also PEEK(43) und PEEK(44).



B. Geben Sie nun im Direktmodus ein:
POKE 43,(PEEK(45)+256*PEEK(46)-2)AND255 (RETURN)
POKE 44,(PEEK(45)+256*PEEK(46) -2/256) (RETURN)

C. Programm B laden. Dann die notierten Werte wieder in die Speicherzellen 43 und 44 POKEn.

Fertig – beide Programme sind jetzt gemeinsam im Speicher.

Bei diesem Vorgang ist zu beachten, daß zuerst das Programm mit den niedrigeren Zeilennummern geladen wird (Programm A) – ansonsten kann es zu Schwierigkeiten kommen beziehungsweise das MERGEn klappt einfach nicht.

Wir werden deshalb schon sehr bald auf die beste Dateiverwaltungsmethode zu sprechen kommen, die auf dem C64 überhaupt möglich ist: Relative Dateien mit Hilfe des Diskettenlaufwerks 1541.

Würde man auf Disk eine Datei mit 2000 Sätzen anlegen und in dieser dann einen Begriff suchen, der an letzter Stelle der Datei steht, so würde man bei Verwenden unserer bisherigen Suchroutine eine viel größere Suchzeit als 14 Sekunden in Kauf nehmen müssen. Auch eine Programmierung in Assembler würde keine befriedigenden Suchzeiten ergeben, da bei relativen Dateien auf Disk noch ein anderer Faktor eine wesentliche Rolle spielt: Die Zugriffzeit des Floppy-Laufwerks – also die Zeit, die das Disketten-Laufwerk braucht, um auf bestimmte Sätze einer relativen Datei zuzugreifen. Diese Zugriffszeit, die vom Floppy-Laufwerk beansprucht wird, läßt sich auch mit Assembler nicht beeinflussen. Auch Hypra-Load-Fans werden enttäuscht sein, da Hypra-Load wie auch alle anderen ähnlichen Programme (Ausnahme: Parallelübertragung à la SpeedDos etc.) die Zugriffszeit auf relative Dateien in keinster Weise beeinflussen. Hier auch gleich eine Anmerkung für alle, die ihr altes Betriebssystem durch ein Betriebssystem mit Hypra-Load ersetzt haben (beziehungsweise Umschaltplatine im Computer haben): Hypra-Load beansprucht den Datasetten-Puffer von 828-1019. Diesen Bereich werde ich ab und zu für kleine MP-Routinen beanspruchen. Es ist daher unbedingt erforderlich, daß Sie dann das alte Betriebssystem im Computer haben beziehungsweise auf dieses umschalten!

```
10 DIM X$(2000)                              <005>
20 X$(2000)="MIKE"                           <114>
30 REM SUCHROUTINE                           <014>
35 TI$="000000":REM STOPPUHR AUF NULL        <110>
40 FOR I=1 TO 2000                           <091>
50 :IF X$(I)="MIKE"THEN PRINT"MIKE STEHT I
   N SATZ";I:PRINT"BENOETIGTE ZEIT:";TI$     <165>
60 NEXT I                                    <144>
```

**Listing 7. Suchvorgang gestoppt**

Nun aber zurück zum Sortieren.

Bevor wir unsere erste Sortierroutine programmieren, müssen wir uns noch einiges an Theorie zu Gemüte führen.

Betrachten wir hierzu noch einmal unsere Früchtedatei. Bisher haben wir immer nur einzelne Begriffe gesucht. Was aber nun, wenn wir aus unserer Früchtedatei alle Früchte heraussuchen wollen, die einer Beerensorte angehören? Wir müßten als Antwort dann ERDBEERE, HIMBEERE und BROMBEERE erhalten. Woher in aller Welt soll unser C64 aber wissen, was Beeren sind beziehungsweise welche Früchte als Beeren bezeichnet werden können?

Die einfachste Lösung wäre die folgende:

Alle Beerenarten in unserer Datei haben die Endung BEERE gemeinsam. Es liegt nun nahe, beim Suchvorgang einfach zu überprüfen, ob die letzten 5 Zeichen jedes Früchtenamens gleich »BEERE« ist; also:

```
IF RIGHT$(D$,5)=X$ THEN
```

wobei X$="BEERE" ist.

Um die Suchroutine (Listing 4) entsprechend zu ändern, müssen Sie lediglich Zeile 130 modifizieren;

```
130 : IF RIGHT$(D$(SA),LEN(X$))=X$THENPRINTD$(SA):
IC=IC+1
```

Die erste Hürde bei der Einführung von Dateiverwaltungsprogrammierung wäre nun geschafft. Sie wissen jetzt, wie man einfache Dateien innerhalb von Basic tabellenartig verwaltet, und Sie haben die einfachste und schlechteste Suchroutine kennengelernt, die es gibt.

## Ohne Sortieren geht es nicht

Bisher haben wir unsere Früchtedatei von Anfang bis Ende durchsucht, um schließlich eine bestimmte Frucht zu finden oder zu erfahren, daß sie in der Datei gar nicht beziehungsweise mehrmals auftritt. Das ging bisher auch ziemlich

```
30000 REM  QUICKSORT ----------------------
      ----------------------------------
      --                                  <087>
30001 DIM LG(100),RG(100):Z=0:LG(1)=1:RG(1
      )=SZ                                <177>
30010 Z=Z+1:IF LG(Z)>=RG(Z)THEN 30120     <166>
30020 X=LG(Z):Y=RG(Z)                     <011>
30030 VG$=D$(INT((X+Y)/2))                <188>
30040 IF X>Y THEN 30100                   <205>
30050 IF D$(X)<VG$THEN X=X+1:GOTO 30050   <214>
30060 IF D$(Y)>VG$THEN Y=Y-1:GOTO 30060   <064>
30070 IF X>Y THEN 30100                   <235>
30080 S$=D$(X):D$(X)=D$(Y):D$(Y)=S$       <130>
30090 X=X+1:Y=Y-1:GOTO 30040              <099>
30100 RG(Z+1)=Y:LG(Z+1)=LG(Z):GOSUB 30010 <184>
30110 LG(Z+1)=X:RG(Z+1)=RG(Z):GOSUB 30010 <104>
30120 Z=Z-1:RETURN                        <161>
```

**Listing 8. Der legendäre Quicksort**

schnell, in Bruchteilen von Sekunden. Was aber nun, wenn wir riesige Dateien mit mehreren tausend Sätzen verwalten wollen?

Die »Schwäche« unserer bisherigen Suchroutine wird bei dem Demo-Programm in Listing 7 besonders deutlich:

Hier wird in einer Datei mit 2000 Sätzen der Name »Mike« gesucht, der unglücklicherweise an letzter Stelle der Datei steht. Das Programm sucht 14 Sekunden, bis es den Namen endlich gefunden hat. Ein kleiner Tip am Rande: SYS 65499 bewirkt das gleiche wie der Befehl TIS="000000".

Diese Zeit ist sogar noch akzeptabel, wenn man bedenkt, daß sich auf unsere Früchtedatei keine andere vernünftige beziehungsweise schnellere Suchroutine als die obig vorgestellte anwenden läßt.

Warum? Ganz einfach – vernünftige Suchroutinen lassen sich nur bei Dateien anwenden, die nach einem bestimmten Kriterium sortiert sind (zum Beispiel nach der alphabetischen Grundordnung).

Wir müssen unsere Früchtedatei also zunächst sortieren, bevor wir eine bessere Suchroutine programmieren können.

»Warum programmieren wir die obige Suchroutine nicht einfach in Maschinensprache?« könnte jetzt einer rufen! Er hätte auch recht – die obige Routine wäre in Assembler derartig schnell, daß wir uns über die Programmierung von Suchroutinen keine Gedanken machen müßten.

Es muß also noch andere Gründe geben, warum ich das Zeitproblem nicht einfach mit einer Maschinensprache-Routine unter den Tisch kehre und stattdessen einen Umweg über die Programmierung von Sortieralgorithmen mache. Und der ist wie folgt: Sie müssen sich lediglich vor Augen halten, daß wir es im Moment nur mit einfachen Dateien innerhalb von Basic zu tun haben. Diese Dateien sind zwar sehr gut zu handhaben, aber ihre Verwaltungskapazität ist zu stark eingeschränkt – nämlich auf 38 KByte Basic-RAM beim C 64.

Wenn Sie das Programm nun starten und als Suchbegriff »BEEREN« eingeben, so gibt die Suchroutine alle Sätze aus, die mit »BEEREN« enden – also alle Beerensorten. Genausogut können Sie als Suchbegriff »COLA« eingeben und erhalten dann alle in der Datei vorkommenden Cola-Sorten aufgelistet. Diese Suchmethode funktioniert im Moment zwar ganz gut, ist jedoch viel zu sehr eingeschränkt, da sich Dateien nur in ganz seltenen Fällen nach ihren Satzendungen sortieren lassen. Angenommen, Sie möchten eine Datei programmieren, in der Computerprogramme verwaltet werden (zum Beispiel für ein Softwarehaus). Man müßte dann nach verschiedenen Kategorien einteilen: Zum Beispiel Programmart, Preislage, Computertyp etc.

Ein solches Programm für einen Software-Händler müßte folgende Bedingungen aufweisen:

1. Feststellen, ob ein bestimmtes Programm geführt wird.
2. Datei nach bestimmten Kriterien sortieren:
   - Programmart
   - Preislage etc.
3. Lagerbestand kontrollieren – also wieviele Exemplare eines Programmes noch vorrätig sind etc.

Man müßte die Dateiverwaltung in Form einer Tabelle aufbauen, die so aussehen könnte:

| 1. Schlüssel Programmname | 2. Schlüssel Art | 3. Schlüssel Preis | 4. Schlüssel Vorrat |
|---|---|---|---|
| vizawrite | 1 | 200.00 | 24 |
| wordpro | 1 | 1155.50 | 13 |
| space invaders | 2 | 39.00 | 67 |
| jumpman | 2 | 249.00 | 34 |
| simons basic | 3 | 99.00 | 65 |
| g-basic | 3 | 3140.00 | 45 |
| gordon saga | 4 | 39.00 | 89 |
| deadline | 4 | 99.00 | 34 |

Die einzelnen Spalten beziehungsweise Kategorien einer solchen Tabelle/Datei bezeichnet man als Schlüssel. Wenn ich die Software-Datei zum Beispiel nach dem 3. Schlüssel sortiere, so sortiere ich nach dem Kriterium Preis. Beim 1. Schlüssel – dem Programmnamen – würde ich alphabetisch sortieren.

Bevor man nun zur Programmierung übergeht, muß man sich noch einige andere Gedanken über den Aufbau der Tabelle machen. Man muß so zum Beispiel genau festlegen, aus wieviel Zeichen der Programmname maximal bestehen darf. Auch für alle anderen »Schlüssel« muß genau festgelegt werden, welche Länge sie einnehmen dürfen. Betrachten wir einmal den zweiten Schlüssel: Um zwischen 9 Programmarten unterscheiden zu können, benötige ich ein Zeichen Platz. Lege ich jedoch 2 Zeichen als maximale Länge fest, so kann ich bis zu 99 (also zweistellige Zahlen) Arten unterscheiden. Unsinnig wäre es sicherlich, 4 Stellen zu reservieren, da man wohl kaum seine Programme in 9999 verschiedene Arten einteilen will. Auch wenn ich momentan nur 9 Programmarten unterscheide (also nur 1 Zeichen Platz benötige), entscheide ich mich für 2 Zeichen Reservierung, damit meine Datei möglichst ausbaufähig bleibt.

Für den 2. Schlüssel muß ich noch eine weitere Tabelle anlegen:

1 = Textverarbeitungsprogramm
2 = Actionsspiel
3 = Basic-Erweiterung
4 = Abenteuerspiel

Nach einigen Überlegungen lege ich schließlich folgendes fest:

| | |
|---|---|
| Programmnamenlänge | : max. 15 Zeichen |
| 2. Schlüssel | : max. 2 Zeichen |
| 3. Schlüssel | : max. 6 Zeichen |
| 4. Schlüssel | : max. 2 Zeichen |



Die Tabelle, wie sie später im Computer sein soll, ist damit festgelegt wie in Bild 1.

| 1. SCHLÜSSEL | 2. SCHLÜSSEL | 3. SCHLÜSSEL | 4. SCHLÜSSEL |
|---|---|---|---|
| VIZAWRITE | 01 | 200.00 | 24 |
| WORDPRO | 01 | 155.50 | 13 |
| SPACE INVADERS | 02 | 39.00 | 67 |
| JUMPMAN | 02 | 249.00 | 34 |
| SIMONS BASIC | 03 | 99.00 | 65 |
| G-BASIC | 03 | 140.00 | 45 |
| GORDON SAGA | 04 | 39.00 | 89 |
| DEADLINE | 04 | 99.00 | 34 |

**Bild 1. Aufbau der Software-Datei**

Lassen Sie mich noch einmal kurz zu unserer Früchtedatei zurückkehren. Diese Datei läßt sich besonders einfach sortieren, da sie aus nur einem Schlüssel besteht (dem Fruchtnamen).

Zum Sortieren braucht man einen möglichst guten Sortier-Algorithmus – am besten in Form eines Basic-Unterprogramms, das Sie später einfach in Ihrem eigenen Programm einbauen können.

Ich halte es nicht für sinnvoll, in diesem Kurs lange und breit auf die Funktion der einzelnen Sortieralgorithmen einzugehen beziehungsweise diese vorzustellen. Der schnellste Sortieralgorithmus ist der Quicksort. Ihn wollen wir immer dann im Kurs anwenden, wenn irgendetwas sortiert werden muß. Wer sich für die Grundlagen der Sortieralgorithmen interessiert, den möchte ich auf den Kurs »Effektives Programmieren« hinweisen, der im Jahre 1985 in der Zeitschrift 64'er veröffentlicht worden ist. Dort erfahren Sie alles Wis-

senswerte über die Programmierung und Funktion dieser Algorithmen.

Listing 8 zeigt das Unterprogramm zum Quicksort-Algorithmus.

Der Quicksort ist wie bereits gesagt der schnellste, aber auch der am schwierigsten verständliche Algorithmus zum Sortieren. Für den Verlauf dieses Kurses ist es lediglich notwendig, daß Sie wissen, wie das Quicksort-Unterprogramm angewendet wird beziehungsweise welche Parameter man ihm übergeben muß.

Beachten Sie bitte die beiden DIMs in Zeile 30001. Wenn Sie innerhalb eines Programmes den Quicksort mehrmals verwenden wollen, also das Unterprogramm mit GOSUB30000 aufrufen, müssen Sie diese beiden DIMs an einer anderen Stelle im Programm (am besten am Anfang) unterbringen, daß es nicht zu einem REDIM'D ARRAY ERROR kommt.

Das Unterprogramm benötigt folgende Parameter:
- das Feld, das sortiert werden soll, muß D$(1)–D$(?) sein. Sie können im Unterprogramm natürlich auch D$ durch etwas anderes ersetzen!
- In der Variablen SZ steht die Anzahl der Sätze der Datei/des Feldes.
- LG(1) und RG(1) geben Anfang und Ende des Bereiches des Feldes, das sortiert werden soll, an.

Ist LG(1)=1 und RG(1)=SZ, so heißt das, daß das gesamte Feld D$ sortiert werden soll. Ist LG(1)=10 und RG(1)=40, so wird vom 10ten bis zum 40sten Satz sortiert etc.

Das Beispielprogramm in Listing 9 zeigt, wie sich unsere Früchtedatei mit Hilfe des Quicksort-Unterprogramms alphabetisch sortieren läßt.

An diesem Programm braucht nicht viel erklärt werden – einfach Abtippen und RUN.

Interessant ist vielleicht, was mit unserem Satz Nummer 9 der Früchtedatei passiert ist (der leere Satz). Er steht nach dem Sortiervorgang an erster Stelle der Datei. Wenn Sie möchten, dann können Sie leere Sätze als eine Art NOPs betrachten – einfache Platzhalter, in die man später noch etwas einbauen kann. Bei Assembler-Programmierung sind solche Platzhalter unentbehrlich, bei Dateiverwaltung jedoch relativ überflüssig, da gute Dateiverwaltungen Befehle enthalten, innerhalb einer Datei automatisch zwischen zwei Sätzen einen Leerraum zu schaffen. Vor dem Sortieren bietet es sich folglich an, die leeren Sätze aus der Datei zu eliminieren.

Man durchläuft die gesamte Datei hierbei mit einer Schleife und schreibt immer dann, wenn man auf einen leeren Satz findet, den letzten Satz dieser Datei in den leeren Satz. Danach wird SZ, die Variable, die angibt, wieviele Sätze die Datei enthält, um den Wert 1 erniedrigt.

Das geht so wie in Listing 10.

Sie müssen zum Aufruf des Unterprogramms lediglich Listing 9 um folgende Zeile ergänzen:

```
11 GOSUB20000:REM LEERE SAETZE WEG
```

Nun werden vor dem Sortiervorgang alle leeren Sätze aus der Datei entfernt. Wir sind nun in der Lage, jede (sich komplett im Computer befindliche) Datei zu sortieren. Daher eröffnen sich uns auch völlig neue Suchroutinen:

Die binäre und die index-sequentielle Suchmethode.

Für unsere folgenden Experimente gibt die Früchtedatei nicht mehr genug her. Wir wenden uns deshalb nun der Software-Datei zu, die über mehrere Schlüssel verfügt.

Zunächst wird die Datei wieder in einem Unterprogramm ab 10000 definiert, wie es Listing 11 zeigt.

Eigentlich haben wir unsere Dateien bisher immer doppelt im Speicher stehen: einmal innerhalb von DATA-Zeilen und zum anderen in einem dimensionierten Stringfeld. Wir gehen bisher also ziemlich verschwenderisch mit dem geringen Speicher unseres C64 um (gering in Anbetracht der Tatsache, daß es heute schon Computer zu Homecomputer-

```
10 GOSUB 10000:REM TABELLE DEFFINIEREN          <247>
20 PRINT"{CLR}TABELLE VOR DEM SORTIEREN:        <042>
30 FOR I=1 TO SZ                                <235>
40 :PRINT I,D$(I)                               <118>
50 NEXT I                                       <134>
60 PRINT"{DOWN}[RETURN]":POKE 198,0:WAIT 1
   98,1                                         <220>
65 PRINT"BITTE WARTEN!"                         <000>
70 GOSUB 30000:REM D$(1)-D$(SZ) SORTIEREN       <009>
80 PRINT"{CLR}TABELLE NACH DEM SORTIEREN:       <218>
90 FOR I=1 TO SZ                                <039>
100 :PRINT I,D$(I)                              <178>
110 NEXT I                                      <194>
120 END                                         <122>
```

**Listing 9. Steuerprogramm zu Quicksort**

```
20000 REM LEERE SAETZE AUS EINER DATEI ELI
      MINIEREN ----------------------------
      --                                        <088>
20020 I=0: REM ZAEHLER AUF NULL                 <039>
20030 I=I+1:IF I=SZ THEN RETURN                 <163>
20040 :IF D$(I)=""THEN D$(I)=D$(SZ):SZ=SZ-
      1:GOTO 40                                 <222>
20050 GOTO 20030                                <146>
```

**Listing 10. Aufräumen einer Datei**

```
10000 REM FRUECHTE-DATEI DEFFINIEREN -----
      ------------------------------------
      --                                        <026>
10010 SZ=8       :REM ANZAHL DER SAETZE         <038>
10020 DIM D$(SZ):REM STRING-FELD                <163>
10030 FOR I=1 TO SZ                             <073>
10040 : READ D$(I)                              <073>
10050 NEXT I                                    <228>
10100 DATA"VIZAWRITE{6SPACE}01200.0024"         <086>
10105 DATA"WORDPRO{8SPACE}01155.5013"           <036>
10110 DATA"SPACE INVADERS 02 39.0067"           <175>
10115 DATA"JUMPMAN{8SPACE}02 49.0034"           <111>
10120 DATA"SIMONS BASIC{3SPACE}03 99.0065"      <230>
10125 DATA"G-BASIC{8SPACE}03140.0045"           <186>
10130 DATA"GORDON SAGA{4SPACE}04 39.0089"       <093>
10135 DATA"DEADLINE{7SPACE}04 99.0034"          <132>
10140 RETURN                                    <036>
```

**Listing 11. Dateidefinition einmal anders**

```
10 REM D$(1)-D$(SZ) ALS SEQ-FILE ---------
   -------------------------------------        <061>
20 GOSUB 10000: REM TABELLE DEFFINIEREN         <001>
30 OPEN 2,8,2,"DATEN-FILE,S,W"                  <242>
40 FOR I=1 TO SZ                                <245>
50 :PRINT#2,D$(I)                               <007>
60 NEXT I                                       <144>
65 CLOSE 2                                      <084>
70 PRINT"*** FERTIG ***"                        <079>
80 END                                          <082>
```

**Listing 12. So speichert man die Datei auf Disk**

```
10000 REM FRUECHTE-DATEI DEFFINIEREN -----
      ------------------------------------
      --                                        <026>
10010 SZ=8       :REM ANZAHL DER SAETZE         <038>
10020 DIM D$(SZ):REM STRING-FELD                <163>
10030 OPEN 2,8,2,"DATEN-FILE,S,R"               <206>
10040 FOR I=1 TO SZ                             <083>
10050 :INPUT#2,D$(I)                            <197>
10060 NEXT I                                    <238>
10070 CLOSE 2                                   <183>
10080 RETURN                                    <232>
```

**Listing 13. Lesen einer Datei von Disk**

Preisen gibt, die auf 512 KByte RAM aufgestockt werden können!). Es wird also höchste Zeit, sich um eine bessere Unterbringung unserer Tabellen zu bemühen.

In der Tat – sind die DATA-Zeilen erst einmal in ein Stringfeld eingelesen, werden sie nicht mehr weiter benötigt und blockieren lediglich freien Speicherplatz.

Oh, soeben habe ich einen kleinen Fehler in Listing 11 entdeckt: Die REM-Zeile 10000 darf nicht länger Früchte-Datei heißen, sondern muß jetzt Software-Datei lauten. Wir müssen

Zeile 10000 deshalb folgendermaßen abändern:

```
10000 REM SOFTWARE-DATEI DEFINIEREN
```

Vielleicht werden sich einige (mit gutem Recht) wundern, warum ich hier lange und breit auf diesen lächerlichen »Fehler« eingehe und nicht einfach mein Manuskript verbessere........ kein Kommentar, denn jetzt kommt wieder ein kleiner (aber äußerst interessanter) Tip am Rande:

## Ärger beim »Replace«'n

Selbstverständlich gehe ich jetzt davon aus, daß Sie alle Listings, die Sie im Rahmen dieses Kurses abtippen, auf Diskette speichern, um diese später wieder verwenden zu können. Nehmen wir einmal an, daß Sie auch das falsche Listing 11 bereits gespeichert haben. Worauf ich hinaus will, ist das folgende: Wenn Sie das »falsche« Listing bereits gespeichert haben, so werden Sie es sicherlich verbessern und anschließend mit dem Befehl

```
SAVE"@:LISTING 11",8
```

wieder speichern. Wenn Ihnen nicht klar ist, was das »@« im Filename soll, so sind Sie noch nicht »reif« genug für den folgenden Tip – lesen Sie schleunigst Ihr Bedienerhandbuch zur VC 1541!

Wer die Anweisung kennt und schon des öfteren angewandt hat, hat sicherlich auch schon unerklärbare Probleme damit gehabt. So kann es zum Beispiel vorkommen, daß der Befehl einfach nicht funktioniert und die rote LED am Laufwerk ähnlich einer extrem miserablen 1-Kanal-Lichtorgel im 4/4-Takt zu blinken beginnt. Die Erklärung dieses Fehlers im Betriebssystem ist sehr kompliziert. Für den Anwender ist nur eines wichtig: Niemals den @ verwenden! Sicherer ist es, mit »Scratch« zu löschen und dann noch einmal zu speichern. Also: Niemals SAVE "@..."

Nun aber schnell zurück zu unserem aktuellen Problem. Laden Sie bitte Listing 11, falls Sie es nicht mehr im Speicher haben. Nach Eingabe von CLR und PRINT FRE(0)+65000 erhalten Sie die Antwort, daß 37928 Byte Basic RAM frei sind. Nach Starten des Programmes mit GOSUB 10000 im Direktmodus und Eingabe von PRINT FRE(0) erhalten Sie die Antwort, daß jetzt nur noch 37880 Bytes Basic RAM frei sind. Es wird also Basic-RAM benötigt, um ein Stringfeld zu füllen. Es macht bei unserem Programm zwar nur ein paar Bytes aus, aber unsere Tabelle ist ja extrem klein. Bei großen Dateien kann ein sehr großer Unterschied vorkommen.

Es bieten sich nun mehr Lösungsmöglichkeiten an.

Man kann so zum Beispiel darauf verzichten, die DATAs in ein Stringfeld einzulesen. Man benötigt dann eine modifizierte RESTORE-Routine. Im Basic des C 64 wird mittels RESTORE der DATA-Zeiger einfach auf die erste DATA-Zeile zurückgesetzt. Unter modifizierte RESTORE-Funktion versteht man nun die Möglichkeit, den DATA-Zeiger auf eine bestimmte DATA-Zeile zu setzen – RESTORE(X), wobei X für die DATA-Zeile steht, auf die der Zeiger gesetzt werden soll. Von dieser Möglichkeit, die sich hervorragend zum Programmieren von Adventures anbietet, die komplett im Basic-RAM stehen (worauf Datasetten-Benutzer angewiesen sind), habe ich in meinem ersten Adventure-Programmierkurs im zweiten 64'er Sonderheft regen Gebrauch gemacht. Das entsprechende Maschinenprogramm für die modifizierte RESTORE-Routine finden Sie in diesem Heft.

Für Dateiverwaltung, KI-Programme und disketten-orientierte Adventures ist die RESTORE-Methode jedoch absolut ungeeignet. Die Tabelle kann innerhalb von Basic nicht ausgebaut werden – und gerade dies ist bei Dateiverwaltung und selbstlernenden KI-Programmen unvermeidlich (schon gut, es gibt Basic-Zeilengeneratoren, aber wir wollen uns mit effektvolleren Methoden vertraut machen!).

Eine wesentlich effektvollere Methode besteht darin, die DATA-Zeilen durch sequentielle Dateien zu ersetzen.

Wir ergänzen deshalb Listing 11 nun um ein Steuerprogramm, das D$(1) bis D$(SZ) als sequentielle Datei auf Disk ablegt (Listing 12).

Die Software-Datei ist jetzt unter dem Filenamen »Daten-File« als SEQ-File auf Diskette gespeichert. Unter einem sequentiellen File versteht man eine Datenkette, die auf Diskette gespeichert ist – also ein String beziehungsweise Element nach dem anderen. Eine so gespeicherte Tabelle kann ganz einfach wieder in den Computer in ein dimensioniertes Feld geladen werden (Listing 13).

Die hier vorgestellten Beispielprogramme sollen uns zunächst genügen. Es gibt noch einige weitere, wesentlich bessere Methoden, Dateien und Tabellen als sequentielle Files zu speichern. Tatsächlich ist es auch nicht besonders sinnvoll, jeden einzelnen Satz einer Tabelle mit mehreren Schlüsseln als einfachen String zu speichern. Wie ich Ihnen bereits in der Einleitung zum Kurs mitteilte, möchte ich das Thema Dateiverwaltung speziell auf die Bereiche KI und fortgeschrittene Adventure-Programmierung beziehen. SEQ-Files sind auch mit der Datasette möglich. Beim Programmieren von Adventures und KI sind sie jedoch nur äußerst begrenzt sinnvoll einsetzbar. Wie es der Name schon sagt, werden sequentielle Dateien kettenförmig gespeichert. Es ist somit nicht möglich, einen bestimmten Teil der Datei so mir nichts, dir nichts zu verändern beziehungsweise in der Mitte der Datei eine Lücke zu schaffen, um nachträglich etwas einzutragen. Zur Bearbeitung muß eine SEQ-Datei immer komplett in den Computer geladen werden. Dann, nach Beendigung der Arbeit, wird sie wieder auf Diskette gespeichert. Eine sequentielle Datei läßt sich nur dann effektiv modifizieren, wenn sie so klein (relativ betrachtet) ist, daß sie auf einmal in den Speicher des C 64 paßt. Dadurch ist die maximale Kapazität einer solchen Datei sehr eingeschränkt.

Für kleinere Dateien sind sequentielle Dateien jedoch empfehlenswert, da sie sehr einfach zu handhaben sind. Für die späteren KI-Programme und Adventures werden SEQ-Files nur eine kleine Nebenrolle spielen. Ich möchte daher nicht auf alle Möglichkeiten der sequentiellen Datenspeicherung eingehen, zumal zu diesem Thema auch genügend geschrieben worden ist.

Jetzt wird es aber höchste Zeit, einige weitere Experimente zum Sortieren und Suchen an unserer neuen Datei durchzuführen. Als Ausgangsprogramm verwenden Sie bitte immer Listing 11. Für unsere Beispiele lohnt es sich nicht, die Datei jedesmal als SEQ-file laden zu müssen.

## Sortieren mit dem Quicksort

Wie wir anhand des Demoprogramms zur Suchzeit (Listing 7) festgestellt haben, ist unsere erste Suchmethode – Datei vom ersten bis zum Zielelement durchsuchen – viel zu langsam, wenn eine große Datei vorliegt, und das gesuchte Element ziemlich am Ende der Datei steht. Wir waren uns auch einig, daß bessere Suchmethoden nur dann sinnvoll sind, wenn die Datei alphabetisch beziehungsweise nach einem anderen Kriterium sortiert ist. Besonders deutlich kommen die verschiedenen Sortierkriterien anhand einer Personendatei zum Vorschein. Wenn ich zum Beispiel wissen will, ob eine bestimmte Person in der Datei aufgeführt ist, so muß ich die Datei alphabetisch ordnen (Namen), um ein schnelles Suchprogramm schreiben zu können. Will ich jedoch wissen, wer an einem bestimmten Tag alles Geburtstag hat, so muß ich die Datei nach den Geburtstagen sortieren. Das Sortieren von Geburtstagen ist schon etwas komplizierter als das Sortieren nach dem Alphabet.

Beispiel:

```
Tobi    21.11.66
Borgsi  30.05.67
Olaf    20.03.67
```

Als String lassen sich die Geburtstage nicht so ohne weiteres sortieren. Ein kleiner Trick hilft jedoch:

```
Tobi    661121
Borgsi  670530
Olaf    670320
```

Die Geburtstage werden durch einfaches »Umdrehen« in sechsstellige Zahlen verwandelt. Die höchste Zahl kommt jetzt dem jüngsten der drei zu, die niedrigste dem ältesten.

Jetzt können die »Geburtstage« einfach als Strings mit dem Quicksort oder einem anderen Sortieralgorithmus sortiert werden.

Zurück zur Software-Datei. Wie wir bereits festgestellt haben, sind verbesserte Suchmethoden nur dann möglich, wenn die Datei irgendwie (nach irgendeinem Schlüssel) sortiert ist. Bei unserer Software-Datei bietet sich der 1. Schlüssel – also der Programmname – an. Listing 14 zeigt, wie die Datei mit dem Quicksort alphabetisch geordnet wird (1. Schlüssel).

Bitte beachten Sie unbedingt die Änderungen des Quicksort-Unterprogramms in den Zeilen 30030, 30050 und 30060. Diese Zeilen müssen immer dann modifiziert werden, wenn nicht ein ganzer String, sondern nur ein Teilbereich des Strings als Sortierargument gelten soll. Beim 1. Schlüssel unserer Software-Datei sind dies die jeweils ersten 15 Zeichen vom Gesamtstring.

Wir stellen uns nun folgende Aufgabe: Nach Eingabe des Programmnamens gibt das Programm Art, Preis und Lagerbestand zu dem Programm aus.

Eine einfache Lösung wäre es wieder, die ganze Datei nach dem Programmnamen zu durchsuchen. Da unsere Programmnamen jedoch alphabetisch geordnet sind, können wir mit einer anderen Suchmethode arbeiten, mit der binären Suchmethode.

## Die binäre Suchmethode



Betrachten wir einmal den geordneten Schlüssel der Software-Datei:

```
1 DEADLINE
2 G-BASIC
3 GORDON SAGA
4 JUMPMAN
5 SIMONS BASIC
6 SPACE INVADERS
7 VIZAWRITE
8 WORDPRO
```

Wir wollen nun das Programm GORDON SAGA suchen, um festzustellen, in welchem Satz (Satznummer) es steht.

Wir setzen zunächst einen Zeiger auf den mittleren Satz der Datei. Die Satzanzahl SZ beträgt 8. Der mittlere Satz SA errechnet sich als SA=INT(SZ/2). Im ersten Durchgang wird der Zeiger also auf 4 gesetzt.

Nun untersuchen wir, ob unser Suchbegriff GORDON SAGA gleich Satz 4, kleiner als Satz 4 oder größer als Satz 4 ist. Wir kommen zu dem Ergebnis, daß der Suchsatz kleiner als Satz 4 ist. Der Suchsatz muß also in der ersten Hälfte der Datei stehen (im Bereich Satz 1 – Satz 3).

Wir halbieren den Restbereich und haben als neue Mitte Satz 2. GORDON SAGA ist jedoch größer als Satz 2. Unser Suchsatz kann also nur noch im Bereich von Satz 3 – Satz 3 stehen – also in Satz 3. Damit haben wir ihn gefunden.

Die binäre Suche besteht also darin, Bereiche nach oben beziehungsweise nach unten hin zu halbieren, bis der Suchsatz gefunden ist. Das Schema in Bild 2 macht die binäre Suchmethode noch viel deutlicher. SU$ ist der Begriff (String), der gesucht werden soll. D$(SA) ist der Inhalt (String) des Satzes mit der Nummer SA beziehungsweise des SAten Satzes der Datei.

```
10 GOSUB 10000:REM DATEI DEFFINIEREN              <103>
20 PRINT"DATEI UNGEORDNET:"                       <035>
30 FOR I=1 TO SZ:PRINT D$(I):NEXT I               <120>
40 GOSUB 30000:REM 1.SCHLUESSEL SORTIEREN         <078>
50 PRINT"DATEI NACH 1.SCHLUESSEL SORTIERT:
   "                                              <137>
60 FOR I=1 TO SZ:PRINT D$(I):NEXT I               <150>
70 END                                            <072>
10000 REM FRUECHTE-DATEI DEFFINIEREN -----
      -----------------------------------
      --                                          <026>
10010 SZ=8      :REM ANZAHL DER SAETZE            <038>
10020 DIM D$(SZ):REM STRING-FELD                  <163>
10030 FOR I=1 TO SZ                               <073>
10040 : READ D$(I)                                <073>
10050 NEXT I                                      <228>
10100 DATA"VIZAWRITE{6SPACE}01200.0024"           <086>
10105 DATA"WORDPRO{8SPACE}01155.5013"             <036>
10110 DATA"SPACE INVADERS 02 39.0067"             <175>
10115 DATA"JUMPMAN{8SPACE}02 49.0034"             <111>
10120 DATA"SIMONS BASIC{3SPACE}03 99.0065"        <230>
10125 DATA"G-BASIC{8SPACE}03140.0045"             <186>
10130 DATA"GORDON SAGA{4SPACE}04 39.0089"         <093>
10135 DATA"DEADLINE{7SPACE}04 99.0034"            <132>
10140 RETURN                                      <036>
30000 REM  QUICKSORT --------------------
      -----------------------------------
      --                                          <087>
30001 DIM LG(100),RG(100):Z=0:LG(1)=1:RG(1
      )=SZ                                        <177>
30010 Z=Z+1:IF LG(Z)>=RG(Z)THEN 30120             <166>
30020 X=LG(Z):Y=RG(Z)                             <011>
30030 VG$=LEFT$(D$(INT((X+Y)/2)),15)              <071>
30040 IF X>Y THEN 30100                           <205>
30050 IF LEFT$(D$(X),15)<VG$THEN X=X+1:GOT
      O 30050                                     <015>
30060 IF LEFT$(D$(Y),15)>VG$THEN Y=Y-1:GOT
      O 30060                                     <034>
30070 IF X>Y THEN 30100                           <235>
30080 S$=D$(X):D$(X)=D$(Y):D$(Y)=S$               <130>
30090 X=X+1:Y=Y-1:GOTO 30040                      <099>
30100 RG(Z+1)=Y:LG(Z+1)=LG(Z):GOSUB 30010         <184>
30110 LG(Z+1)=X:RG(Z+1)=RG(Z):GOSUB 30010         <104>
30120 Z=Z-1:RETURN                                <161>
```

Listing 14. Sortieren einer Datei

```
1110 : IF SU$=MID$(D$(SA),1,LEN(SU$))THEN
     PRINT"SU$ STEHT IN SATZ ";SA:GOTO 200
     0                                            <160>
1120 : IF SU$<MID$(D$(SA),1,LEN(SU$))THEN
     SA=SA-(2↑(N-1)):GOTO 1200                    <250>
2000 REM SU$ IST GEFUNDEN - AUSWERTUNG ---
     -----------------------------------
     -                                            <163>
2010 PRINT"{CLR,RVSON}";LEFT$(D$(SA),15);"
     {RVOFF}":PRINT                               <221>
2020 PRINT"PROGRAMMART: ";                        <201>
2030 I=VAL(MID$(D$(SA),16,2))                     <176>
2040 IF I=1 THEN PRINT"TEXTVERARBEITUNG"          <235>
2050 IF I=2 THEN PRINT"ACTIONSPIEL"               <055>
2060 IF I=3 THEN PRINT"BASIC-ERWEITERUNG"         <246>
2070 IF I=4 THEN PRINT"ADVENTURE"                 <171>
2080 PRINT"PREIS: ";MID$(D$(SA),18,6);" DM
     "                                            <111>
2085 PRINT"BESTAND: ";VAL(RIGHT$(D$(SA),2)
     );"STUECK"                                   <056>
2090 GOTO 1000                                    <254>
```

Listing 14b. Ergänzung zu Listing 14.

Man setzt also jeweils einen Zeiger in die Mitte des Bereiches, in dem sich der gesuchte Begriff befinden muß. Am Anfang des Suchvorgangs ist dies logischerweise die Mitte der gesamten Datei. Danach wird untersucht, ob er Begriff gefunden wurde beziehungsweise ob der sich im Bereich unterhalb oder überhalb des Zeigers befindet. Stellt man wie in unserem Bild beim 1. Vergleich fest, daß sich der gesuchte Begriff unterhalb des Zeigers befindet, so wird der neue Zeiger auf die Mitte des unteren Bereiches etc. gesetzt. Was jedoch, wenn ein Suchbegriff überhaupt nicht in einer Datei vorkommt?

Ganz einfach, man kann bei einer Datei mit bekannter Satzanzahl SZ genau berechnen, wieviele Vergleiche maximal

gemacht werden müssen, um einen Begriff zu finden, der in der Datei enthalten ist. Die binäre Suchmethode beruht schließlich auf dem Halbieren von Bereichen.

Gehen wir wieder einmal von der Software-Datei aus. Sie enthält genau 8 Elemente. Nach dem ersten Suchversuch wird die Datei halbiert. Es kommen dann nur noch 4 Elemente zur Auswahl, in deren Mitte der zweite Zeiger positioniert ist. Nach dem zweiten Versuch kommen nur noch 2 Elemente in Frage und spätestens nach dem 4. Versuch schließlich nur noch eine.

Spätestens beim 4. Versuch (Halbierung) muß ein Suchbegriff somit in der Software-Datei gefunden werden – andernfalls kommt er nicht vor.

Die Formel, mit der sich die maximal nötige Suchvorgangs-Anzahl berechnen läßt, lautet so:

N=INT(LOG(SZ)/LOG(2))+1

Nach N-Versuchen muß ein Begriff also gefunden sein, wenn er in einer Datei mit SZ-Sätzen enthalten ist.

Am besten läßt sich die binäre Suchroutine anhand eines entsprechenden Unterprogrammes (Listing 15) erklären.

Das Unterprogramm setzt folgendes voraus:

Das Feld, in dem gesucht werden soll, steht in D$(0)–D$(SZ), wobei SZ die Nummer des höchsten Satzes beziehungsweise die Anzahl der Sätze darstellt.

**Dokumentation:**

**1010** Eingabe des Suchbegriffs SU$.

**1020** Berechnung der maximal nötigen Suchvorgänge: N

**1030** SA, der Zeiger auf die Datei wird zu Beginn des Suchvorgangs in die Mitte der Datei gesetzt. Warum berechnet sich die Mitte nicht einfach mit SA=SZ/2? Ganz einfach: Zum Halbieren »der Restbereiche« verwenden wir Potenzen von 2 (wird gleich erläutert). Probleme würde es geben, wenn eine Datei aus einer ungeraden Anzahl von Sätzen bestünde. Man könnte dann schlecht halbieren. Eine »ungerade Datei« einfach mit einem »Pseudozusatzsatz« zu versehen, würde auch nichts bringen – früher oder später kämen beim Halbieren ungerade SA-Positionen zustande – und die darf's nicht geben. Die Formel $SA=(2xa^{N})/2$ gewährleistet jedoch, daß Nmal halbiert werden kann, wobei die Restbereiche stets aus einer geraden Anzahl von Elementen bestehen.

**1040** Nach jedem Suchvorgang (Halbierung) wird die Variable N um 1 erniedrigt. Ist sie kleiner als 0, so kommt der Suchbegriff in der Datei nicht vor.

**1050 – 1100** Halbierungsvorgang

**1060** Prüfen, ob D$(SA) gleich SU$ ist. Wenn ja, so ist der Begriff gefunden worden und die Suche wird abgebrochen.

**1070** Wenn SU$ kleiner als D$(SA) ist, so heißt dies, daß sich SU$ (falls es vorkommt) nur in dem Bereich über dem Zeiger SA befinden kann. Der Zeiger muß also auf die Mitte dieses Bereiches gesetzt werden. Das Versetzen geschieht mit der Potenzfunktion: $SA=SA-(2*a^{(N-1)})$.

Dadurch, daß N nach jedem Suchbegriff um 1 verkleinert wird, werden die Restbereiche und somit die Verschiebungsdistanz des Zeigers auch immer kleiner.

**1080–1090** Diese Zeilen treten dann in Kraft, wenn SU$ größer als D$(SA) ist. Die Verschiebungsformel lautet dann: $SA=SA+(2*^{(N-1)})$, da der Zeiger jetzt in den unteren Bereich verschoben werden muß. Um einen Überlauf zu verhindern, ist Zeile 1090 nötig. In Zeile 1030 idealisierten wir unsere Datei so, daß nie ungerade Zeigerwerte auftreten können. Die Datei wird also größer angenommen, als sie eigentlich ist. Das Vergrößern der Datei ist die einzige Möglichkeit zum Idealisieren. Würde man sie verkleinern, so gingen Sätze verloren beziehungsweise würden beim Suchen außer acht gelassen werden. Die Zeile 1080 ruft solange rekursiv auf, bis der Zeiger SA sich wieder im erlaubten Bereich von 0 bis SZ befindet – falls er diesen Bereich verlassen hat. Es ist erlaubt, den Zeiger außerhalb des erlaubten Bereiches zu positionieren; es ist jedoch klar, daß ein Suchbegriff außer-

```
1000 REM BINAERE SUCHROUTINE -------------
     ------------------------------------
     -                                    <046>
1005 INPUT SU$  :REM SUCHBEGRIFF          <209>
1010 N=INT(LOG(SZ)/LOG(2))+1
                :REM MAXIMAL-FORMEL       <137>
1040 SA=(2↑N)/2:REM MITTE DER GES.DAT     <071>
1050 N=N-1      :REM 1.ABFRAGE            <086>
1100 :                                    <060>
1110 : IF SU$=D$(SA)THEN PRINT"SU$ STEHT I
     N SATZ ";SA:END:REM SU$ GEFUNDEN     <033>
1120 : IF SU$<D$(SA)THEN SA=SA-(2↑(N-1)):G
     OTO 1200       :REM ''HALBIEREN''    <030>
1125 : IF SA+(2↑(N-1))>SZ AND N>-1 THEN N=
     N-1:GOTO 1125      :REM ''UEBERLAUF'' <147>
1130 : SA=SA+(2↑(N-1))
                :REM ''HALBIEREN''        <246>
1200 :                                    <160>
1210 :N=N-1     :REM NAECHST. HALBIEREN   <149>
1220 :IF N<0 THEN PRINT"SU$ GIBT'S NICHT!
     IN 1220":END    :REM SU$ NICHT IN DAT
     EI                                   <159>
1225 PRINT"SA=";SA;" N=";N
                :REM UNNOETIG !!          <036>
1230 :GOTO 1100 :REM NAECHST. SUCH-VERS.  <123>
```

**Listing 15. Die binäre Suchmethode**

halb der Datei gefunden werden kann. Auch kann es nur im Fall SU$ > D$(SA) zu einer Positionierung außerhalb des Datenbereiches kommen.

**1110** Nächsten Suchvorgang vorbereiten

**1120** Ist N kleiner als 0, so befindet sich SU$ nicht in der Datei.

**1130** kann jederzeit entfernt werden.

Dies hört sich alles schön und gut an, ich weiß jedoch, daß sich viele schwer tun, wenn sie das erste Mal mit binären Suchsystemen konfrontiert werden. Ich rate Ihnen deshalb, falls Sie Schwierigkeiten haben, sich das obige Programm

1. Vergleich SU$ > D$(X)

3. Vergleich SU$ > D$(X)

4. Vergleich SU$ = D$(X) gefunden!

2. Vergleich SU$ < D$(X)

D$(1)–D$(SZ) Sortierte Datei mit SZ-Sätzen.

Satznummer X

**Bild 2. So funktioniert das binäre Suchen, bei dem der zu durchsuchende Bereich »halbiert« wird**

einmal Schritt für Schritt durchzudenken. Nehmen Sie gegebenenfalls ein Blatt Papier und eine kleine Datei her und schreiben Sie den gesamten Programmablauf (wo sich der Zeiger nach jedem Suchvorgang befindet) in Form von einfachen Schemen auf. Hat man es erstmal verstanden, dann ist es ganz einfach.

Bitte ergänzen Sie jetzt Listing 14 mit Listing 15. Außerdem sind folgende Änderungen vorzunehmen:

Zeile 70 löschen.

Ergänzen beziehungsweise ändern Sie auch die Zeilen, wie in Listing 14b angegeben.

Die binäre Suchmethode ist sehr schnell. Um eine Datei von 1000 Sätzen nach einem Begriff zu durchsuchen, sind maximal 10 Halbierungen notwendig, um festzustellen, in welchem Satz der Begriff steht, beziehungsweise daß er nicht vorkommt.

Erhöhen kann man die Suchgeschwindigkeit noch durch Einsatz von index-sequentiellen Dateien, wie wir sie bei Parser 4.0 kennenlernen werden.

Das Programm, das Sie jetzt im Computer haben, simuliert eine einfache Dateiverwaltung für Software.

Damit haben Sie den ersten Einstieg in die Dateiverwaltung auch schon geschafft. Wir können uns nun Dateien auf Disk zuwenden, die für die Programmierung von Adventures und KI ein großes Hilfsmittel darstellen.

## Deutscher Zeichensatz im Commodore 64

Bevor ich mich der Programmierung von relativen Dateien widme, möchte ich schnell noch dieses Kapitel einfügen. Wer professionelle deutsche Text-Adventure schreiben will (und das können Sie am Ende dieses Kurses!), sollte keineswegs auf einen deutschen Zeichensatz verzichten.

Auch bei allen anderen Programmen mit deutschen Texten macht ein deutscher Zeichensatz einen besseren Eindruck. Natürlich gehen bei einer Zeichensatzänderung immer ein paar KByte-RAM flöten (der Zeichensatz muß schließlich vom ROM ins RAM kopiert werden), aufgrund guter Programmiertechniken, die sehr platzsparend sind, können wir diesen Verlust jedoch getrost verschmerzen.

Auch bei relativen Dateien bringt ein neuer Zeichensatz erhebliche Vorteile, wie wir schon bald sehen werden.

Zunächst müssen wir festlegen, wo im RAM der neue Zeichensatz liegen soll.

Bild 3 zeigt den Speicherbelegungsplan des C 64, wie er Ihnen wahrscheinlich bereits von anderen Kursen aus dem 64'er-Magazin her bekannt ist. Uns interessieren nur der linierte Bereich und das Basic-RAM. Im Laufe des Kurses werden wir den gesamten linierten RAM-Bereich hernehmen (für Maschinenprogramme, Dateien etc.). Den Basic-RAM versuche ich weitgehend für Ihre eigenen Ideen und Routinen zur freien Verfügung zu halten. Sie sollten sich nicht wundern, warum bestimmte Routinen (zum Beispiel Zeichensatz) in bestimmte Speicherbereiche geschrieben werden, denn alle Routinen sollen sich am Ende des Kurses bei vollausgelastetem RAM miteinander vertragen. Grübeln Sie also nicht lange nach, wenn ich eine Routine in einen zunächst ungeeignet anmutenden RAM-Bereich schreibe.

Unseren deutschen Zeichensatz bringen wir im Bereich 57344 - 61440 unter. Da der VIC jedoch nur ganze RAM-Blöcke von 16 KByte verwalten kann, müssen wir auch den Videobereich (Bildschirm) verschieben. Ihn legen wir nach 52223 - 53248. Bitte beachten Sie, daß hierbei auch die Sprite-Pointer (2040 - 2047) nach 53239 - 53246 verschoben werden (Bild 4)!



**Bild 4. Hier bringen wir den deutschen Zeichensatz unter**

Bei dem Listing 16 handelt es sich um einen menüfreundlichen Zeichensatzeditor, der speziell für den Kurs abgerichtet ist (also für RAM-Bereich 57344-61440).

**Dokumentation zum Listing 16**

Der Zeichensatzeditor ist lediglich ein Utility zum Kurs. Ich beschränke mich daher auf das Wesentliche.

Interessant ist im Prinzip nur das Abändern des Listings zum Erstellen für Zeichensätze in andere RAM-Bereiche als die Vorgeschlagenen. Ich bin mir jedoch ziemlich sicher, daß der Zeichensatz im Bereich von 57344 - 61440 sehr gut untergebracht ist, wie ich Ihnen schon bald zeigen werde.

Um den Editor für andere Bereiche umzuschrauben, müssen lediglich folgende Zeilen abgewandelt werden:

55 IZ = Startadresse im RAM (ist momentan 57344)
122 siehe Listing
124 siehe Listing
201 VIDEO = 52223
1030 und 1040 siehe Listing
2010 siehe Listing

Es müssen also nur Anfangs- und Endadressen vom ZS-Bereich und gegebenenfalls vom Video modifiziert werden.

**Bedienungsanleitung**

Nach dem Programmstart meldet sich das Hauptmenü mit drei Auswahlmöglichkeiten:



**Bild 3. So sieht die Speicherbelegung des C 64 aus**

CHR$ (64)

CHR$ (94)

CHR$ (95)

CHR$ (173)

CHR$ (174)

CHR$ (176)

CHR$ (181)

CHR$ (182)

**Bild 5. So sehen die deutschen Zeichen aus**

CHR$ (184)

CHR$ (185)

**Geänderte Zeichen**

| | | | | | |
|---|---|---|---|---|---|
| ä | = [€] + [a] | Ä | = [€] + [Z] | B | = [€] + [S] |
| ö | = [€] + [o] | Ö | = [€] + [L] | II | = [@] |
| ü | = [€] + [u] | Ü | = [€] + [J] | : | = [↑] |
| | | | | / | = [←] |

NEUEN Z.SATZ ERSTELLEN: Durch diese Wahl wird der Zeichensatz vom ROM in den RAM-Bereich ab 57344 kopiert.



ALTEN Z.SATZ ÄNDERN: Ein bereits bestehender Zeichensatz kann geladen werden, um weiter verändert zu werden.

DEUTSCHEN Z.SATZ GENERIEREN: Der Editor erstellt hier automatisch einen deutschen Zeichensatz, den Sie anschließend speichern können. Bild 5 zeigt Ihnen, welche Zeichen wie geändert wurden. Ich bin mir sicher, daß Sie sich schnell an die neue Tastaturbelegung gewöhnen werden. Daß auch Zeichen wie ″, und : neu definiert wurden, hat seinen Grund: den erfahren Sie im nächsten Kapitel!

Nach der Wahl von einem der drei Hauptmenüpunkten gelangen Sie ins nächste Menü.

```
0 REM ******************************         <012>
1 REM  ZEICHENSATZ-EDITOR (C) 1985           <146>
2 REM ******************************         <014>
3 IF I=1 THEN 100                            <221>
4 VIDEO=52223:REM START BILDSCHIRM           <012>
5 POKE 53280,0:POKE 53281,0:PRINT"{YELLOW}
  "                                          <235>
10 REM MODIFIZIERTE PEEK-FUNKTION -------    <054>
11 FOR I=704 TO 751:READ X:POKE I,X:NEXT:P
   OKE 785,192:POKE 786,2:REM USR-VEKTOR     <009>
12 DATA 165,20,72,165,21,72,32,247,183,165
   ,1,72,165,21,201,208,144,7,201,224        <079>
13 DATA 176,3,169,49,44,169,52,120,133,1,1
   60,0,177,20,168,104,133,1,88,104,133      <228>
14 DATA 21,104,133,20,76,162,179             <191>
30 REM MENUE ---------------------------     <009>
31 PRINT"{CLR,WHITE,RVSON}ZEICHENSATZ-EDIT
   OR I{RVOFF}"                              <037>
32 PRINT"{DOWN}VON MICHAEL NICKLES.          <006>
34 PRINT"{3DOWN,5SPACE,RVSON}N{RVOFF}EUEN
   ZEICHENSATZ ERSTELLEN"                    <216>
35 PRINT"{DOWN,5SPACE,RVSON}A{RVOFF}LTEN Z
   EICHENSATZ AENDERN{2SPACE}"               <076>
36 PRINT"{DOWN,5SPACE,RVSON}D{RVOFF}EUTSCH
   EN ZEICHENSATZ GENERIEREN{2SPACE}"        <243>
37 GET I$:IF I$<>"N"AND I$<>"A"AND I$<>"D"
   THEN 37                                   <009>
38 IF I$="N"THEN 50                          <171>
39 IF I$="D"THEN 50                          <167>
40 REM DEUTSCHEN ZEICHENSATZ LADEN -----     <178>
41 INPUT"{3DOWN}FILENAME ";I$                <206>
42 IF I<>1 THEN I=1:LOAD I$,8,1              <074>
44 POKE 648,204:POKE 53272,56:POKE 56576,0
   :PRINT"{CLR}"                             <069>
50 REM Z.S. VOM ROM INS RAM AB 57344 ---     <247>
51 PRINT"{DOWN}BITTE WARTEN!"                <080>
53 POKE 56334,PEEK(56334)AND 254:POKE 1,PE
   EK(1)AND 251: REM INTERRUPT AUS           <161>
55 IZ=57344                                  <001>
60 FOR I=53248 TO 57344                      <033>
62 :POKE IZ,PEEK(I)                          <139>
64 :IZ=IZ+1                                  <248>
66 NEXT I                                    <150>
70 POKE 1,PEEK(1)OR 4:POKE 56334,PEEK(5633
   4)OR 1                                    <223>
80 IF I$="D"THEN 3000                        <209>
100 REM START ------------------------       <226>
105 POKE 648,204:POKE 53272,56:POKE 56576,
    0:PRINT"{CLR,WHITE}";                    <169>
106 PRINT"BITTE WAEHLEN SIE:                 <223>
110 PRINT"{2DOWN,5SPACE}GROSS-GRAFIK Z.SAT
```

**Listing 16. Verzichten Sie in deutschen Adventures nicht auf den deutschen Zeichensatz**

```
     Z{2SPACE,RVSON}1{RVOFF}"                     <021>
115 PRINT"{DOWN,5SPACE}GROSS-KLEIN{2SPACE}
     Z.SATZ{2SPACE,RVSON}2{RVOFF}"                <026>
116 PRINT"{DOWN,5SPACE}Z.SATZ ABSPEICHERN{
     3SPACE,RVSON}3{RVOFF}                        <210>
118 PRINT"{DOWN,5SPACE}CRAZY CHARACTERS{5S
     PACE,RVSON}4{RVOFF}                          <080>
120 GET I$:IF I$<>"1"AND I$<>"2"AND I$<>"3
     "AND I$<>"4"THEN 120                         <091>
122 IF I$="1"THEN AN=57344        :PRINT CHR
     $(142)                                       <132>
124 IF I$="2"THEN AN=57344+256*8:PRINT CHR
     $(14)                                        <051>
125 IF I$="3"THEN 1000                            <109>
127 IF I$="4"THEN 2000                            <244>
130 PRINT"{2DOWN}WELCHES ZEICHEN WOLLEN SI
     E AENDERN?"                                  <082>
131 PRINT"ANGABE BITTE IM BILDSCHIRMCODE!         <190>
132 INPUT"{DOWN}ZEICHEN ";NR:Z=NR                 <234>
200 REM MASKE AUSGEBEN -----------------          <013>
201 VIDEO=52223                                   <169>
205 PRINT"{CLR}ZEICHEN: ":POKE VI+10,Z            <144>
210 Z=AN+Z*8                                      <160>
229 PRINT"{2DOWN,4SPACE}";                        <180>
230 FOR I1=Z TO Z+7                               <018>
235 PRINT I1;"{3SPACE}";                          <233>
240 :B=USR(I1)                                    <060>
245 : FOR I2=7 TO 0 STEP-1                        <036>
250 :    IF B AND 2↑I2 THEN PRINT"*";:GOTO
     260                                          <166>
255 :    PRINT".";                                <191>
260 : NEXT I2                                     <220>
270 :PRINT"{3SPACE}"B:PRINT"{4SPACE}";            <021>
275 NEXT I1                                       <124>
300 REM EDITOR-PROGRAMM ----------------          <045>
310 XX=VI+135:XP=PEEK(XX):POKE XX,32              <130>
330 JOY=56320:REM PORT 2                          <032>
331 TA=203:T1=653                                 <206>
335 IF PEEK(JOY)=127 AND PEEK(TA)=64 THEN
     W=0                                          <140>
340 IF PEEK(JOY)=126 OR PEEK(TA)=7 AND PEE
     K(T1)=1 THEN W=-40                           <054>
342 IF PEEK(JOY)=125 OR PEEK(TA)=7 AND PEE
     K(T1)=0 THEN W=+40                           <023>
344 IF PEEK(JOY)=123 OR PEEK(TA)=2 AND PEE
     K(T1)=1 THEN W=-1                            <244>
346 IF PEEK(JOY)=119 OR PEEK(TA)=2 AND PEE
     K(T1)=0 THEN W=+1                            <061>
350 IF PEEK(JOY)>111 AND PEEK(TA)<>60 THEN
     360                                          <201>
352 IF XP=46 THEN XP=42:GOTO 355                  <078>
354 IF XP=42 THEN XP=46:GOTO 355                  <206>
355 POKE XX,XP:IF PEEK(JOY)<=111 OR PEEK(T
     A)=60 THEN 355                               <204>
360 :IF PEEK(TA)=1 THEN POKE XX,XP:GOTO 40
     0                                            <084>
370 POKE XX,XP                                    <220>
375 XX=XX+W:IF PEEK(XX)=32 THEN XX=XX-W           <020>
380 XP=PEEK(XX):POKE XX,32                        <206>
390 GOTO 330                                      <128>
400 REM NEUES ZEICHEN BERECHNEN                   <026>
405 PRINT"{DOWN}BITTE WARTEN!"                    <180>
410 XX=VI+135                                     <198>
430 FOR I1=XX TO XX+7*40 STEP 40                  <179>
435 :IM=0:IZ=7                                    <186>
445 : FOR I2=I1 TO I1+7                           <149>
450 :    IF RV=0 THEN IF PEEK(I2)=42 THEN I
     M=IM+2↑IZ                                    <155>
451 :    IF RV=1 THEN IF PEEK(I2)=46 THEN I
     M=IM+2↑IZ                                    <036>
455 :    IZ=IZ-1                                  <130>
460 : NEXT I2                                     <164>
470 :POKE Z,IM:Z=Z+1                              <248>
475 NEXT I1                                       <068>
476 IF RV=1 THEN RV=0:GOTO 100                    <041>
479 IF NR>127 THEN 100                            <015>
480 PRINT"REVERS-AENDERUNG ERWUENSCHT? (J/
     N)                                           <210>
482 GET I$:IF I$<>"J"AND I$<>"N"THEN 482          <197>
484 IF I$="N"THEN 100                             <094>
485 Z=AN+(NR+128)*8:RV=1:GOTO 400                 <013>
490 GOTO 100                                      <164>
1000 REM Z.SATZ ABSPEICHERN ------------          <196>
1000 REM SPEICHERN AUF DISK -----------           <092>
1005 PRINT"{CLR,RVSON}ZEICHENSATZ AUF DISK
      SPEICHERN{RVOFF}":POKE 198,0                <178>
1006 INPUT"{2DOWN}FILENAME";I$                    <206>
1010 PRINT"{DOWN}BITTE DISKETTE EINLEGEN!         <153>
1020 PRINT"[RETURN]":POKE 198,0:WAIT 198,1        <000>
1030 AA=57344                                     <240>
1040 EA=61439                                     <051>
1050 OPEN 1,8,1,I$                                <227>
1060 HB=INT(AA/256):LB=AA-HB*256                  <006>
1070 PRINT#1,CHR$(LB);CHR$(HB);                   <179>
1080 FOR I=AA TO EA                               <076>
1090 :PRINT#1,CHR$(USR(I));                       <001>
1100 NEXT I                                       <168>
1110 CLOSE 1                                      <105>
1120 PRINT"{3DOWN}*** FERTIG ***":RUN             <251>
2000 REM CRAZY CHARACTERS --------------          <203>
2010 FOR I=57344 TO 61439                         <079>
2020 :X=USR(I):M=0                                <213>
2030 : FOR I1=0 TO 7                              <128>
2040 :  IF X AND 2↑I1 THEN M=M+2↑(7-I1)           <044>
2050 : NEXT I1                                    <200>
2060 :POKE I,M                                    <044>
2070 :NEXT I                                      <182>
2080 GOTO 100                                     <099>
3000 REM DEUTSCHEN ZEICHENSATZ AUTOMATISCH
      GENERIEREN -------------------------
      -                                           <243>
3010 DATA 60264,102,24,60,102,126,102,102,
      0 :REM GROSSES AE                           <124>
3015 DATA 60336,102,60,102,102,102,102,60,
      0  :REM GROSSES OE                          <143>
3020 DATA 60328,102,0,102,102,102,102,60,0
      :REM GROSSES UE                             <202>
3025 DATA 60288,102,0,60,6,62,102,62,0
      :REM KLEINES AE                             <091>
3030 DATA 60360,0,102,0,60,102,102,60,0
      :REM KLEINES OE                             <068>
3035 DATA 60352,0,102,0,102,102,102,62,0
      :REM KLEINES UE                             <048>
3040 DATA 60272,60,102,102,108,110,102,108
      ,96:REM SCHARFES S                          <124>
3045 DATA 59392,102,102,102,0,0,0,0,0
      :REM ANFUEHRUNGSZEICHEN                     <042>
3050 DATA 59640,0,0,0,0,0,24,24,48
      :REM KOMMA                                  <006>
3055 DATA 59632,0,0,24,0,0,24,0,0
      :REM DOPPELPUNKT                            <218>
3060 DATA-1                                       <148>
3100 READ X:IF X=-1 THEN 100                      <212>
3110 FOR I=X TO X+7                               <049>
3120 : READ A:POKE I,A     :REM NORMAL            <066>
3130 :      POKE I+2048,A:REM REVERS              <110>
3140 NEXT I                                       <176>
3150 GOTO 3100                                    <074>
```

**Listing 16. Ein einfacher Zeichensatzeditor**

```
programm : rueckwaerts       c400 c4bb
--------------------------------------
c400 : a9 c0 8d 88 02 20 44 e5   7c
c408 : a2 15 a0 c4 78 8e 14 03   48
c410 : 8c 15 03 58 60 a2 00 a0   4f
c418 : 27 bd 00 c0 99 00 cc bd   7e
c420 : 28 c0 99 28 cc bd 50 c0   91
c428 : 99 50 cc bd 78 c0 99 78   b9
c430 : cc bd a0 c0 99 a0 cc bd   68
c438 : c8 c0 99 c8 cc bd f0 c0   e0
c440 : 99 f0 cc bd 18 c1 99 18   62
c448 : cd bd 40 c1 99 40 cd bd   8a
c450 : 68 c1 99 68 cd bd 90 c1   9d
c458 : 99 90 cd bd b8 c1 99 b8   d6
c460 : cd bd e0 c1 99 e0 cd bd   cf
c468 : 08 c2 99 08 ce bd 30 c2   5a
c470 : 99 30 ce bd 58 c2 99 58   3f
c478 : ce bd 80 c2 99 80 ce bd   f2
c480 : a8 c2 99 a8 ce bd d0 c2   a8
c488 : 99 d0 ce bd f8 c2 99 f8   f3
c490 : ce bd 20 c3 99 20 cf bd   13
c498 : 48 c3 99 48 cf bd 70 c3   65
c4a0 : 99 70 cf bd 98 c3 99 98   5c
c4a8 : cf bd c0 c3 99 c0 cf e8   af
c4b0 : 88 e0 28 f0 03 4c 19 c4   51
c4b8 : 4c 31 ea 00 00 00 00 00   58
```

**Listing 17. Erster Teil des Überraschungs-Programms**

Sie haben nun folgende Möglichkeiten:

**1: GROSS-GRAFIK Z.SATZ:** Dies müssen Sie wählen, wenn Sie ein Zeichen des GROSS/GRAFIK-Zeichensatzes ändern wollen.

**2: GROSS-KLEIN Z.SATZ:** Wie oben, jedoch GROSS/KLEIN-Zeichensatz.

Nach Wahl von Punkt 1 oder 2 müssen Sie den Bildschirmcode des Zeichens angeben, das Sie ändern wollen (Tabelle siehe C64-Handbuch). Nun kann das Zeichen mittels Joystick oder Cursor-Tasten + SPACE editiert werden. Sind Sie mit der Änderung fertig, so muß die RETURN-Taste gedrückt werden. Sie werden dann gefragt, ob auch das entsprechende Revers-Zeichen geändert werden soll. Dann gelangen Sie wieder zurück ins Menü.

**3: Z.SATZ Speichern:** Der neue Zeichensatz wird auf Disk gespeichert.

**4: Crazy Characters:** Eine Spezialfunktion des Editors (benötigt viel Rechenzeit!). Speichern Sie den mit Crazy-Characters erzeugten Zeichensatz einmal unter dem Filenamen »CRAZY« ab. Wenn Sie nun das Listing17 und 18 eingeben und starten, werden Sie schnell verstehen, warum ein Spiegel zum unentbehrlichen Programmierhilfsmittel werden kann!

Bitte erzeugen Sie nun mittels dem Menüpunkt »DEUTSCHEN Z.SATZ GENERIEREN« einen deutschen Zeichensatz und speichern Sie diesen unter dem Filenamen »D.Z.SATZ« auf Disk – wir werden ihn später benötigen!

**Laden des Zeichensatzes**

So wird der deutsche Zeichensatz innerhalb von Basic-Programmen geladen;

```
10 IFI=0THENI=1:LOAD"D.Z.SATZ",8,1
20 POKE648,204:POKE53272,56:POKE56576,0
30 END
```

Ist ein neuer Zeichensatz aktiviert und ein Programm wird mittels RUN/STOP + RESTORE unterbrochen, so stürzt das Programm ab, der Cursor ist verschwunden. Der Cursor ist allerdings noch da – Sie können ihn momentan nur nicht sehen! Geben Sie nun bitte »blind« folgendes im Direktmodus ein:

```
POKE 648,204     (RETURN)
POKE 53272,56    (RETURN)
POKE 56576,0     (RETURN)
```

Der Cursor ist wieder da! Am besten, Sie schreiben sich diese POKEs gleich auf einen Aufkleber und bringen diesen am C64 an.

## Relative Dateien auf der 1541

Um es gleich zu sagen: Dies ist KEIN Kurs für Disketten-Laufwerke. Ich werde in diesem Kapitel nicht erklären, wie das Floppy-Laufwerk 1541 seine Daten speichert, wie es in Ihrem RAM/ROM aussieht oder gar wie man den Schreib-Lesekopf im $^4/_4$-Takt hin- und herschlagen läßt (als 4. Musikstimme, wie es bei einigen verrückten Programmen schon vorgekommen sein soll). Es geht mir in diesem Kapitel nur darum, daß Sie lernen, spielend leicht mit relativen Dateien umzugehen. Durch den Umgang mit solchen Dateien eröffnen sich völlig neue Programmiermöglichkeiten. Wer erst einmal mit relativen Dateien gearbeitet hat, wird schon bald nicht mehr auf sie verzichten wollen. In der Tat, erst durch relative Dateien ist es möglich, die Speicherkapazität der 1541 voll und ganz auszureizen. Im Bedienungshandbuch für das Diskettenlaufwerk wird kaum auf relative Dateien eingegangen. Auch die großen Bücher über Floppy-Laufwerke geben nicht alle Tips und Tricks zum Umgang mit relativen Dateien preis, sondern handeln alles ganz simpel und logisch ab. Der Anfänger schaut jedoch spätestens dann dumm aus der Wäsche (auch mir ging dies anfangs so), wenn ein FILE DATA ERROR auftaucht oder Satzteile in einer relativen Datei beim Einlesen plötzlich nicht mehr auftauchen. Im Prinzip ist der Umgang mit relativen Dateien jedoch sehr einfach, wenn man erst mal durchgeblickt hat; es gibt nur sehr wenige Kommandos, die nötig sind, um mit relativen Dateien umzugehen: positionieren, lesen und schreiben.

Was ist eine relative Datei?

Am ehesten kann man eine relative Datei mit einem dimensionierten Stringfeld auf Disk vergleichen. Ein Stringfeld im Computer vom Format X$(1)–X$(N) entspricht in etwa einer relativen Datei mit N-Sätzen. Genauso wie ein Stringfeld erst einmal mittels DIM definiert werden muß, muß eine relative Datei erst einmal auf Disk angelegt werden, bevor man mit ihr arbeiten kann.

## Anlegen einer relativen Datei

Beim Öffnen einer relativen Datei kommt es lediglich auf die Angabe von drei Parametern an: Dateiname, Satzlänge und Satzanzahl der Datei. Listing 19 zeigt, wie eine relative Datei eröffnet wird.

Und so geht das vor sich:

| | |
|---|---|
| 20 – 40 | Festlegen der 3 Parameter |
| 50 | Die Satzanzahl SZ wird in Lower und Higher Byte unterteilt. |
| 60 | Öffnen des Floppy-Befehlskanals |
| 70 | Initialisieren der Disk. Dies empfiehlt sich immer dann, wenn eine neue Disk eingelegt beziehungsweise eine Seite gewechselt wird. |
| 80 | Eine relative Datei mit Namen FS und der Satzlänge SL wird geöffnet. |
| 90 | Der relative Datei-Zeiger wird auf den höchsten Satz der Datei positioniert. |
| 100 | In den letzten beziehungsweise höchsten Satz der Datei wird CHR$(255) geschrieben. |
| 110 | Floppy-Datenkanal und relative Datei werden geschlossen. |

```
10 REM CRAZY-LOADER                            <125>
30 IF I2=0 THEN I2=1:LOAD"CRAZY",8,1           <047>
40 IF I2=1 THEN I2=2:POKE 648,204:POKE 532
   72,56:POKE 56576,0                          <082>
60 IF I1=0 THEN I1=1:LOAD"RUECKWAERTS",8,1 <084>
70 IF I1=1 THEN I1=2:SYS 50176:SYS 64764       <100>
```

Listing 18. Zweiter Teil des Überraschungs-Programms

```
 10 REM EROEFFNEN EINER REL.DATEI ---------
    ------------------------------------   <197>
 20 F$="TEST"        :REM DATEINAME        <088>
 30 SL=40            :REM SATZLAENGE       <148>
 40 SZ=50            :REM SATZANZAHL       <172>
 50 HB=INT(SZ/255):LB=SZ-HB*255            <042>
 60 OPEN 15,8,15      :REM FLOPPYKANAL OEFFNE
    N                                      <158>
 70 PRINT#15,"I0"  :REM INITIALISIEREN     <229>
 80 OPEN 1,8,3,F$+",L,"+CHR$(SL)
                     :REM DATEI OEFFNEN    <246>
 90 PRINT#15,"P";CHR$(3);CHR$(LB);CHR$(HB);
    CHR$(1)                                <105>
 100 PRINT#1,CHR$(255)                     <111>
 110 CLOSE 1:CLOSE 15 :REM KANAELE SCHLIESS
     EN                                    <096>
```

Listing 19. Öffnen einer relativen Datei

Nach dem Programmstart kann es eine Weile dauern - dies hängt natürlich von der Größe der Datei ab. Am Ende blinkt die rote LED des Diskettenlaufwerks. Dies zeigt einen Fehler Nr. 50 - »RECORD NOT PRESENT« an. Das Auftauchen dieses Fehlers nach dem Anlegen einer relativen Datei (unmittelbar danach) ist okay. Sie brauchen sich darum keine Sorgen zu machen.

Sie haben bestimmt schon entdeckt, worin sich relative Dateien von dimensionierten Stringfeldern im Speicher unterscheiden: Die Satzlänge, also die maximale Zeichenanzahl, die ein String haben darf, muß genau festgelegt sein. Bei dimensionierten Stringfeldern kann jeder String bis zu 255 Zeichen lang sein. Dies geht natürlich auch bei relativen Dateien, wenn man als maximale Satzlänge SL den Wert 255 angibt. Dies ist jedoch nur selten empfehlenswert, denn: Je kürzer beziehungsweise kleiner die maximal erlaubte Satzlänge SL ist, umso mehr Sätze lassen sich auf einer Disk-Seite unterbringen.

Man sollte sich also am Anfang genau überlegen, wie groß jeder Satz maximal sein muß. Bei der Software-Datei hätte man sich schließlich für die Satzlänge SL=25 entschieden. Man muß auch von Anfang an wissen, wieviele Sätze man benötigt. Dies stimmt nicht ganz, denn man kann relative Dateien prinzipiell auch offen lassen – indem man beim Anlegen der Datei auf den letztmöglichen Satz positioniert und in ihn den Wert 255 schreibt. Davon ist jedoch abzuraten, da offene relative Dateien viel längere Zugriffszeiten erfordern – im Programmablauf also viel mehr Zeit kosten.

Seien Sie beim Anlegen der Dateien stets großzügig. Hat man eine Datei fertig bearbeitet, so kann man leicht eine neue Datei eröffnen, deren Satzlänge und Satzgröße nur so groß ist wie erforderlich. Dann müssen nur noch die Daten aus der ersten Datei in diese Datei übertragen werden. Dies geht ganz einfach mit dem Editor, den Sie in diesem Kapitel noch finden werden.

## Lesen/Beschreiben einer relativen Datei

Bevor man eine relative Datei beschreiben kann, muß natürlich erst einmal eine vorhanden sein. Bitte legen Sie eine leere Disk beziehungsweise Disk mit mindestens 10 freien Blöcken ins Laufwerk und starten Sie dann Listing 19 mittels RUN. Es wird sodann (bitte etwas Geduld) eine relative Datei mit dem Filenamen »TEST« und 50 Sätzen mit einer Länge von je 40 Zeichen angelegt.

Will man auf beziehungsweise von einer relativen Datei schreiben oder lesen, so sind folgende Punkte notwendig: Positionieren, Lesen und Schreiben.

Am deutlichsten wird die, wenn man relative Dateien einmal mit Stringfeldern innerhalb von Basic (X$(1)-X$(N)) vergleicht.

Ein Stringfeld innerhalb von Basic wird einfach mittels
DIM X$(N)
angelegt, wobei N für die Anzahl der Sätze steht. Jeder Satz darf maximal 255 Zeichen umfassen.

Eine relative Datei hingegen muß erst etwas umständlicher angelegt werden. Wie beim Stringfeld muß auch hierbei die Anzahl der Sätze und außerdem auch noch die maximale Länge jedes Satzes angegeben werden.

Will man einen Satz SA von einem Stringfeld lesen beziehungsweise wissen, was im SAten Satz steht, so geht man ganz einfach so vor:
PRINT X$(SA)

Bei relativen Dateien geht dies nicht so einfach. Hier muß erst auf den gewünschten Satz SA positioniert werden, bevor er gelesen werden kann. Also:

Positioniere Zeiger auf Satz SA. Lese Satz SA.

Analog verhält es sich natürlich mit dem Schreiben. Es bietet sich an, die Routinen Schreiben, Lesen und Positionieren als Unterprogramme zu gestalten (Listing 20).

Wie funktionieren die drei Unterprogramme?

Mittels GOSUB 9000 wird der Zeiger auf den Satz SA der relativen Datei positioniert. Die Syntax zum Positionieren des Zeigers ab zum Beispiel Satz 23 einer relativen Datei würde so lauten:

```
SA=23:GOSUB 9000
```

**9010** Hier wird SA in High- und Lowbyte zerlegt. Es ist also möglich, relative Dateien mit bis zu 65000 Sätzen (Satzlänge entsprechend klein) anzulegen.

**9020** Über den Floppy-Befehlskanal wird der Positionierbefehl der Syntax

```
PRINT#15,"P";CHR$(Kanal);CHR$(LOWBYTE);CHR$
(HIGHBYTE) ;CHR$(Zeichen)
```

dem Floppy-Laufwerk mitgeteilt. »Zeichen« gibt an, auf welches beziehungsweise wievieltes Zeichen im Satz positioniert werden soll. Will man den ganzen Satz einlesen (was meist der Fall ist), so muß man auf das erste Satzzeichen positionieren, also CHR$(1). Danach wird der Fehlerkanal abgefragt mittels

```
INPUT#15,A
```

um festzustellen, ob der Satz SA überhaupt vorhanden ist. Nimmt A den Wert 50 an, so bedeutet dies einen RECORD NOT PRESENT ERROR. Dies ist zum Beispiel dann der Fall, wenn wir bei unserer Test-Datei Satz Nummer 70 lesen wollten (die TEST-Datei besteht nämlich nur aus 50 Sätzen).

Mittels GOSUB 9100 wird der Inhalt des Satzes in AS eingelesen, auf den der Zeiger gerade deutet. Will man zum Beispiel wissen,was im 20ten Satz einer Datei steht, so gilt die folgende Syntax:

```
SA=20 : GOSUB 9000 : GOSUB 9100
```

Das Lesen geschieht mittels INPUT # 1,A$, wenn der Satz beziehungsweise die maximal erlaubte Satzlänge der relativen Datei 88 Zeichen nicht überschreitet (Größe des INPUT-Puffers). Wird mit Sätzen gearbeitet, die größer als 88 Zeichen sind, so muß Zeichen für Zeichen mittels GET-Befehl eingelesen werden. Das Lesen funktioniert also wie bei sequentiellen Dateien und hat auch die gleichen Einschränkungen.

Mittels GOSUB 9200 wird der String A$ in den Satz SA der relativen Datei geschrieben. Will man zum Beispiel »SUPERMIKE« in den 34ten Satz einer relativen Datei schreiben, dann geht das so:

```
9000 REM ZEIGER POSITIONIEREN ------------
     -------------------------------------
     -                                     <187>
9001 :                                     <087>
9010 HB%=SA/256:LB%=SA-HB%*256             <068>
9020 PRINT#15,"P";CHR$(3);CHR$(LB%);CHR$(H
     B%);CHR$(1)                           <005>
9030 INPUT#15,A:IF A=50 THEN PRINT"SATZNUM
     MER ZU HOCH"                          <088>
9040 RETURN                                <208>
9100 REM SATZ LESEN ----------------------
     -------------------------------------
     -                                     <178>
9101 :                                     <187>
9110 IF A=50 THEN RETURN                   <082>
9120 INPUT#1,A$                            <116>
9130 RETURN                                <042>
9200 REM SATZ SCHREIBEN ------------------
     -------------------------------------
     -                                     <105>
9201 :                                     <031>
9210 IF A=50 THEN RETURN                   <182>
9220 PRINT#1,A$                            <042>
9230 RETURN                                <144>
```

**Listing 20. Wichtige Routinen für relative Dateien**

```
A$="SUPERMIKE" : SA=34 : GOSUB 9000 : GOSUB 9200
```

Das Listing 21 stellt in Kombination mit den drei Unterprogrammen einen ganz einfachen Editor für relative Dateien dar.

Dieser Editor ist zum Arbeiten mit der Test-Datei gedacht. Eine Dokumentation ist überflüssig, da alles bereits erklärt worden ist.

Schreiben Sie nun einmal ein paar Sätze in die Test-Datei beziehungsweise lesen Sie einmal aus der Test-Datei. Sie werden schnell feststellen, daß beim Lesen oder Schreiben nicht immer auf Disk zugegriffen wird – das Diskettenlaufwerk verhält sich still. Dies hat seinen guten Grund: Die Daten werden nicht augenblicklich auf die Diskette geschrieben, sondern zunächst erst in einem festgelegten RAM-Bereich innerhalb der 1541 gepuffert. Erst zum Beispiel bei Überlauf wird dieser Puffer auf Disk gespeichert. Bei sehr kleinen Dateien wie unserer Test-Datei kann es also passieren, daß sich alles nur im Puffer abspielt. Deshalb ist es unbedingt notwendig, nach Beendigung des gesamten Schreib-Lesevorgangs beziehungsweise dem Arbeiten mit der relativen Datei, die Datei zu schließen – dann wird nämlich der Puffer gespeichert. Ansonsten kann es passieren, daß Daten verloren gehen. Zum Schließen der relativen Datei genügt CLOSE 15. Dies kann auch im Direktmodus durchgeführt werden – zum Beispiel dann, wenn sie mittels RUN/STOP den Editor abgebrochen haben. Also nie vergessen: Nach dem Beenden des Schreibens auf eine relative Datei IMMER die Datei mittels CLOSE 15 (beziehungsweise Floppykanal) schließen. Beim Schließen des Befehlskanals des Diskettenlaufwerks werden übrigens auch alle anderen Kanäle geschlossen. Die Datei muß also nicht extra geschlossen werden – das Schließen des Floppy-Befehlskanals (den man immer mit OPEN 15,8,15 öffnet und mit CLOSE 15 schließt) reicht vollkommen aus.

## Probleme mit relativen Dateien

Bisher hörte sich alles ganz logisch und einfach an, ich möchte es jedoch keinesfalls versäumen, Sie auch auf die Probleme aufmerksam zu machen, die beim Arbeiten mit relativen Dateien immer wieder auftauchen und denen Anfänger oft hilflos gegenüberstehen. Besonders traurig ist die Tatsache, daß bei den folgenden Problemen auch die großen Bücher zur 1541 in keinster Weise helfen!

Besitzer des C128 haben es beim Umgang mit relativen Dateien schon etwas einfacher, denn ihnen steht ein komfortableres Basic als das des C64 zur Verfügung. Es gibt zwar Basic-Erweiterungen für den C64 (zum Beispiel DISK-Basic vom 5. Sonderheft der 64'er), die jedoch viel zu viel Speicher in Anspruch nehmen. Wenn man die einzelnen Probleme der relativen Dateiprogrammierung jedoch kennt, so kommt man auch ganz gut mit den drei Unterprogrammen von Listing 20 zurecht. Nun zu den Problemen beziehungsweise Schwierigkeiten:

### Der Dateizeiger

Was beim Programmieren oft mißachtet wird, ist die Tatsache, daß der Dateizeiger (der auf jeweils einen Satz positioniert werden kann) der relativen Datei nicht ruht – er versetzt sich nach jedem Zugriff automatisch eine weiter.

Beispiel: Satz 5 soll geändert werden.

Zunächst wollen wir wissen, was eigentlich in Satz 5 steht. Also:

```
SA=5 : GOSUB 9000 : GOSUB 9100 : PRINT A$.
```

So schön, so gut, wir wissen jetzt, was in Satz 5 steht. Wo steht jedoch jetzt der Zeiger? Er hat sich automatisch um 1 erhöht. Das heißt, er zeigt jetzt auf Satz 6. Würde man nun erneut GOSUB 9100 : PRINT A$ eingeben, so erhielte man den Inhalt von Satz 6 und der Zeiger würde auf Satz 7 verschoben werden.

Wenn wir also den Inhalt von Satz 5 ändern wollen, so genügt es NICHT

```
A$="neuer Inhalt" : GOSUB 9100
```

einzugeben.

Vielmehr muß es

```
A$="neuer Inhalt" : SA=5 : GOSUB 9000 : GOSUB 9200
```

lauten.

Es muß also vor dem Schreiben erst wieder positioniert werden. Denken Sie also stets daran, vor dem Schreiben immer auf den Satz zu positionieren, in den geschrieben werden soll.

Eine andere Aufgabe: Die Sätze 30 – 40 einer relativen Datei sollen gelesen und mittels PRINT ausgegeben werden. Die Lösung ist ganz einfach:

```
FOR SA=30 TO 40
: GOSUB 9000
: GOSUB 9100
: PRINT A$
NEXT SA
```

```
10 OPEN 15,8,15 :REM FLOPPYKANAL OEFFNEN   <108>
15 PRINT#15,"I0"                           <114>
20 OPEN 1,8,3,"TEST": REM RELATIVE DATEI O
   EFFNEN                                  <027>
100 PRINT"LESEN, SCHREIBEN, ENDE ?"        <070>
110 GET I$:IF I$<>"L"AND I$<>"S"AND I$<>"E
    "THEN 110                              <188>
120 IF I$="L" THEN 1000                    <245>
130 IF I$="S" THEN 2000                    <134>
140 IF I$="E" THEN 3000                    <141>
1000 REM LESEN------------                 <100>
1010 INPUT "SATZNUMMER ";SA                <237>
1020 GOSUB 9000: REM POSITIONIEREN         <244>
1030 GOSUB 9100: REM LESEN                 <066>
1040 PRINT A$  : REM ANZEIGEN              <164>
1050 GOTO 100                              <218>
2000 REM SCHREIBEN--------                 <121>
2010 INPUT "SATZNUMMER ";SA                <221>
2020 INPUT A$ :REM EINGABESATZ             <116>
2030 GOSUB 9000:REM POSITIONIEREN          <238>
2040 GOSUB 9200:REM SCHREIBEN              <252>
2050 GOTO 100                              <202>
3000 REM ENDE-------------                 <041>
3010 CLOSE 1:CLOSE 15                      <128>
3020 END                                   <228>
```

**Listing 21. Einfacher Relativ-Editor**

```
50000 REM INPUT-ROUTINE ---------------    <137>
50001 :                                    <192>
50002 SL=40                                <069>
50005 SL=40                                <072>
50020 PRINT"{DOWN}":BE$="":POKE 198,0:POKE
       211,0:POKE 214,22:SYS 58732:PRINT"{
      YELLOW}@";                           <156>
50030 GET X$:IF PEEK(203)=1 THEN 50120     <226>
50040 IF X$=""THEN 50030                   <171>
50050 IF LEN(BE$)=0 AND ASC(X$) =20 THEN 5
      0030                                 <148>
50060 I=ASC(X$):IF I<32 OR I>133 AND I<159
       THEN IF I<>20 THEN 50030            <131>
50070 IF LEN(BE$)=SL AND I<>20 THEN 50030  <128>
50075 IF X$=" "THEN X$="{RIGHT}"           <012>
50080 BE$=BE$+X$                           <031>
50090 PRINT CHR$(20);X$;"@";               <108>
50100 IF I=20 THEN BE$=LEFT$(BE$,LEN(BE$)-
      2):GOTO 50030                        <101>
50110 GOTO 50030                           <001>
50120 PRINT"{LEFT,SPACE}"                  <227>
50140 RETURN                               <157>
```

**Listing 22. Mit diesem Unterprogramm wird der INPUT-Befehl kräftig aufgemöbelt**

Diese Lösung funktioniert zwar perfekt, es geht jedoch auch viel einfacher und schneller:

```
SA=30 : GOSUB 9000
FOR I=30 TO 40
: GOSUB 9100
: PRINT A$
NEXT I
```

Die zweite Lösung, die vom automatischen Vorrücken des Dateizeigers Nutzen zieht, benötigt nur halb soviel Zeit, um die 10 Sätze auszugeben wie die erste Lösung. Es lohnt sich also stets, das automatische Vorrücken des Zeigers beim Programmieren zu beachten.

**Steuerzeichen**

Versuchen Sie einmal mittels dem Editor in den 10. Satz der TEST-Datei den Satz beziehungsweise den Inhalt »LOLO, PETER UND MIKE GINGEN IN DEN WALD« zu schreiben. Sie werden schnell merken, daß dies nicht geht, da die INPUT-Routine des C64 keine Kommas zuläßt – es kommt ein EXTRA IGNORED. Hier hilft nur eine verbesserte INPUT-Routine, die das Listing 22 darstellt:

Die Routine wird einfach mittels GOSUB 50000 aufgerufen. Im String BE$ steht dann der Eingabesatz. Interessant im Listing ist lediglich Zeile 50020.

Mittels

```
POKE 211,SPALTE:POKE 214,ZEILE:SYS 58732
```

kann der Cursor an eine beliebige Stelle des Bildschirms positioniert werden.

In Zeile 50005 muß mittels der Variablen SL festgelegt werden, wie groß die maximale Länge des Eingabestrings sein darf.

Erweitern Sie bitte den Editor (Listing 21) um Listing 22.

Außerdem muß noch Zeile 2020 so geändert werden, daß die neue INPUT-Routine aufgerufen wird:

```
2020 GOSUB 50000 : A$=BE$
```

Jetzt können Sie »LOLO, PETER UND MIKE IN DEN WALD« in Satz 10 schreiben. Enttäuschung wird jedoch breit, wenn Sie anschließend Satz 10 wieder lesen wollen.

Sie erhalten beim Lesen nur »LOLO«. Alles, was nach dem Komma steht, einschließlich dem Komma ist verschwunden. Das Komma läßt sich innerhalb relativer Dateien also nicht so einfach verwenden, wie wir es gerne hätten. Wir können jedoch auch nicht auf das Komma verzichten, wenn wir Texte schreiben wollen.

Das gleiche Problem taucht bei Verwendung eines Doppelpunktes auf. Ganz zu schweigen von den ewigen Problemen mit den Gänsefüßchen beziehungsweise Anführungszeichen innerhalb von Strings.

Wie bei den meisten Problemen gibt es allerdings auch hier eine einfache Lösung. Der Editor wird einfach um folgende Zeile ergänzt:

```
2035 A$=CHR$(34)+A$+CHR$(34)
```

Bevor A$ abgespeichert wird, wird es also einfach in Anführungszeichen gesetzt. Jetzt können alle Sonderzeichen verwendet werden. Auch beim Einlesen bereiten die beiden Anführungszeichen zu Beginn und am Ende von A$ keine Schwierigkeiten – sie tauchen nicht auf. Die Sache hat wie immer natürlich ihren Haken: Die maximal erlaubte Satzlänge beträgt jetzt nicht mehr 40, sondern nur noch 38 Zeichen, da die Anführungszeichen zwei Zeichen Platz benötigen. Es wird LOLO, PETER UND MIKE also wieder nicht gelingen, endlich in den Wald zu gelangen. Abhilfe würde hier ganz einfach das Anlegen einer neuen TEST-Datei mit 42 Zeichen Satzlänge schaffen.

Hier sind jedoch auch Überlegungen dem Tun vorauszustellen. Nehmen wir einmal an, auf einer Diskette wäre Platz für maximal 1000 Sätze mit je 40 Zeichen. Da wir jedoch 42 brauchen, um Strings mit 40 Zeichen darzustellen, würden wir nur noch zirka 952 Sätze auf der Diskette unterbringen. Unsere jetzige Lösung ist somit zwar einfach, jedoch nicht sehr effektiv, da sie viel zu verschwenderisch mit dem Disk-Speichervolumen umgeht.

Sicherlich werden sich einige von Ihnen daran erinnern, daß ich im letzten Kapitel behauptete, daß ein deutscher Zeichensatz auch enorme Vorteile beim Programmieren von relativen Dateien mit sich bringt. Einige haben sich sicherlich auch schon gewundert, warum ich neben den deutschen Umlauten und »ß« auch noch »"«, »,«, »:« und »;« nochmals definiert habe.

## Komma ist nicht gleich Komma

Wenn wir unseren deutschen Zeichensatz im Computer haben, so können wir am Aussehen des Kommas auf dem Bildschirm nicht unterscheiden, ob es sich um CHR$(44) (dem Originalkomma) oder um CHR$(95) (Pfeil nach links, der als Komma umdefiniert wurde) handelt. Unser Computer hingegen kann es.

Geben Sie im Direktmodus einmal folgendes ein:

PRINT """ und drücken RETURN

Als Antwort erscheint nichts.

Geben Sie nun

PRINT""" ein.

Das mittlere Anführungszeichen müssen Sie allerdings durch Drücken der @-Taste eingeben (also das definierte Anführungszeichen).

Diesmal erscheint als Antwort tatsächlich ein Anführungszeichen. Der umdefinierte Zeichensatz machts möglich! Erweitern Sie den Editor bitte um folgende Zeilen:

```
1 IFI=OTHENI=1:LOAD"D.Z.SATZ",8,1
2 POKE648,204:POKE53272,56:POKE56576,0:
```

Jetzt wird zu Beginn des Programmes der deutsche Zeichensatz (den Sie hoffentlich bereits mit dem Zeicheneditor automatisch erzeugt und als D.Z.SATZ auf Diskette gespeichert haben!) geladen.

Auf Zeile 2035 können wir nun verzichten – wir brauchen die Anführungszeichen links und rechts von A$ nicht mehr. Das heißt, in einer Datei mit Satzlänge von 40 Zeichen können jetzt auch wirklich Sätze mit 40 und nicht mit 38 Zeichen untergebracht werden.

Jetzt können LOLO, PETER UND MIKE auch endlich in den Wald gehen (natürlich ohne das Bier zu vergessen). Auf die Frage, was LOLO, PETER UND MIKE eigentlich überhaupt im Wald wollen beziehungsweise wozu sie das Bier mitgenommen haben, werde ich später noch zu sprechen kommen!

Bei Eingabe von Sätzen dürfen Sie von nun an jedoch nicht vergessen, daß die Tastatur in bezug auf Komma, Doppelpunkt und Anführungszeichen nicht mehr stimmt – wenn Sie Texte für relative Dateien schreiben wollen. Anstelle der Komma-Taste muß jetzt die Pfeil-links-Taste betätigt werden, wenn ein »Pseudokomma« gewünscht wird etc. Sie werden sich sicher schnell mit der neuen Tastaturbelegung und dem deutschen Zeichensatz anfreunden.

Durch die enormen Vorteile, die sich uns jetzt bieten, gibt es wirklich kein Argument mehr, das gegen die Verwendung eines deutschen Zeichensatzes spricht.

Wenn man sich für die Verwendung des deutschen Zeichensatzes anstelle der Lösung mit den beiden Anführungszeichen entscheidet, müssen noch einige Dinge berücksichtigt werden:

**SPACEs**

Spaces – Leerräume am Anfang eines Satzes sind nicht ohne weiteres möglich. Angenommen, wir schreiben in Satz 30 den String »HALLO«. Lesen wir den Satz 30 anschließend, so erhalten wir die Antwort »HALLO«. Die 10 Spaces werden vom Computer zu Beginn des Satzes einfach ignoriert. Es ist jedoch unsinnig, ein neues Space zu definieren, auch wenn dies eine Lösung wäre. Wir verwenden anstelle

von Space (CHR$(32) und CHR$(160)) einfach CHR$(29) – Cursor nach rechts. Hierfür sorgt Zeile 50075 im Editor. Zu beachten ist hierbei, daß, wenn später mit Strings gearbeitet werden soll, die aus einer mit dieser Methode geführten relativen Datei stammen, die man zum Auffinden von Wortzwischenräumen nicht nach » «, sondern nach CHR$(29) suchen muß!!!

Im Gegensatz zu »,«, »:« und »"« kann zum Erzeugen eines Spaces jedoch ruhig die Space-Taste gedrückt werden (und nicht die Cursor-nach-rechts-Taste). Verwirrt? Probieren Sie einfach mal alles am Computer aus, dann wird alles schnell klar!

**Leersätze**

Versuchen Sie doch mal einfach, einen Leersatz zu schreiben. Also einfach RETURN drücken. Soll der Satz wieder gelesen werden, so steigt der Computer aus – er findet nichts mehr und verhält sich still. Es ist nicht möglich, in relative Dateien Leersätze " " zu schreiben, die nicht in Anführungszeichen stehen. Da wir uns jedoch für die modifizierte Zeichensatz-Lösung entschieden haben, müssen wir erneut einen kleinen Kunstgriff machen: Wir schreiben anstelle von Leersätzen einfach Sätze, die mit einem Doppelpunkt beginnen, denn wir haben ja erfahren, daß alles, was nach einem Komma beziehungsweise Doppelpunkt steht, als nichtig interpretiert wird. Wir müssen den Editor nur um folgende Zeile ergänzen:

```
50130 IF BE$="" THEN BE$=":"
```

Damit wäre auch schon das Wichtigste über relative Dateien gesagt. Wer die Vorzüge dieser Dateien erst einmal schätzen gelernt hat, wird sie schon bald für unentbehrlich halten.

**Einschränkung**

Das Disketten-Laufwerk 1541 hat nur drei Datenkanäle zur Verfügung. Eine relative Datei beansprucht zwei dieser Kanäle, das heißt, es kann jeweils nur eine relative Datei geöffnet sein. Will man mit zwei oder mehreren relativen Dateien arbeiten, so muß beachtet werden, daß immer die eine Datei zuerst geschlossen werden muß, wenn man eine andere öffnen will. Die Datenübertragung von einer relativen Datei zu einer anderen ist somit nur mittels eines Puffers möglich.

Neben einer relativen Datei kann auch noch eine sequentielle Datei geöffnet werden, da diese Dateiart nur einen Kanal benötigt. Man kann also maximal drei sequentielle Dateien oder eine relative und eine sequentielle gleichzeitig geöffnet haben.

## Der Relative-Dateien-Editor

Im folgenden stelle ich Ihnen meinen Relative-Dateien-Editor vor, der uns für den Rest des Kurses als unentbehrliches Werkzeug dienen wird.

Der Editor hat sich im Laufe der Monate allmählich aus dem Editor entwickelt. Er ist keineswegs so komfortabel wie ein Textverarbeitungssystem, was bei Relative-Datei-Editoren auch nie der Fall sein wird beim C64. Ich habe großen Wert auf Anwenderfreundlichkeit des Programmes gelegt: Der Editor hat nur eine einzige Menüzeile, von der sich jederzeit alle nur erdenklichen Zwischenmenüs aufrufen lassen. Der Editor eignet sich hervorragend zum Erstellen von Texten von Adventure-Spielen. Er eignet sich auch prima zum Erstellen von Wortschätzen – wichtig für Parsertechnik, wie wir bald sehen werden. Der Editor arbeitet auch zusammen mit dem deutschen Zeichensatz. Am besten ist es, wenn Sie den Editor gemeinsam mit dem Zeichensatzeditor und dem deutschen Zeichensatz auf eine Diskette speichern. Im Laufe des Kurses werde ich Ihnen noch weitere Programme für diese Utility-Disk vorstellen. Zunächst sollten Sie jedoch erst einmal das Listing 23 zum Editor abtippen:

```
0 REM ********************************     <131>
1 REM *     TEXTEDITOR V.III.4        *     <052>
2 REM *          <C> 1985             *     <177>
3 REM *     VON MICHAEL NICKLES       *     <213>
4 REM ********************************     <135>
8 IF IM=1 THEN POKE 648,204:POKE 53272,56:
  POKE 56576,0:PRINT"{CLR}":GOTO 20         <217>
9 POKE 53280,0:POKE 53281,0:PRINT"{CLR,WHI
  TE}";                                     <053>
10 M%=300:DIM M$(M%),M(M%):REM PUFFER       <065>
15 DIM LG(100),RG(100)      :REM Q.SORT     <062>
19 REM ANFANGSMENUE ----------------------
   ---------------------------------------  <094>
20 PRINT"{CLR,RVSON,8SPACE}TEXTEDITOR - VE
   RSION 3{10SPACE,RVOFF}";                 <159>
21 PRINT"{RVSON,4SPACE}<C> 1985 VON MICHAE
   L NICKLES{8SPACE,RVOFF}";                <196>
22 PRINT"{2DOWN,5SPACE}ALTE DATEI EDITIERE
   N{2SPACE}-{2SPACE}'A'                    <080>
23 PRINT"{5SPACE}NEUE DATEI EROEFFNEN{2SPA
   CE}-{2SPACE}'N'                          <188>
24 PRINT"{DOWN,5SPACE}D.Z.SATZ LADEN{8SPAC
   E}-{2SPACE}'D'                           <241>
25 GET I$:IF PEEK(203)=1 THEN I$="A"        <101>
26 IF I$<>"A"AND I$<>"N"AND I$<>"D"THEN 25 <098>
27 IF I$="D"THEN IM=1:LOAD"D.Z.SATZ",8,1    <110>
28 POKE 198,0:INPUT"{2DOWN}DATEINAME{2SPAC
   E}*{3LEFT}";D$:POKE 198,0                <087>
29 INPUT"SATZLAENGE{2SPACE}40{4LEFT}";SL:I
   F SL<1 OR SL>254 THEN PRINT"GEHT NICHT
   !":GOTO 30                               <200>
30 IF I$="A"THEN 40                         <024>
31 INPUT"SATZANZAHL ";SN:HB=INT(SN/255):LB
   =SN-HB*255                               <094>
32 PRINT"BITTE LEERE DISK EINLEGEN <RETURN
   >":POKE 198,0:WAIT 198,1                 <092>
33 OPEN 15,8,15                             <191>
35 PRINT#15,"I0":OPEN 1,8,3,D$+",L,"+CHR$(
   SL)                                      <237>
36 PRINT#15,"P";CHR$(3);CHR$(LB);CHR$(HB);
   CHR$(1)                                  <051>
37 PRINT#1,CHR$(255)                        <048>
38 CLOSE 1:CLOSE 15                         <206>
40 OPEN 15,8,15                             <198>
41 PRINT#15,"I0":OPEN 1,8,3,D$:PRINT        <191>
50 REM ZWISCHENMENUE ---------------------
   ---------------------------------------  <253>
51 PRINT"{RVSON}S / L / V / O / D / I / Z
   / F / ENDE{4SPACE,RVOFF}";               <011>
52 GET I$:IF I$=""THEN 52                   <225>
53 IF I$="E"THEN PRINT"{CLR}";:GOTO 9900    <034>
54 IF I$="V"THEN PRINT"{CLR}";:GOTO 6000    <130>
55 IF I$="O"THEN PRINT"{CLR}";:GOTO 10000   <251>
56 IF I$="D"THEN PRINT"{CLR}";:GOTO 7000    <125>
57 IF I$="I"THEN PRINT"{CLR}";:GOTO 30000   <254>
58 IF I$="Z"THEN PRINT"{CLR}";:GOTO 8000    <140>
59 IF I$="F"THEN PRINT"{CLR}";:GOTO 61000   <137>
60 IF I$<>"S"AND I$<>"L"THEN 52             <183>
61 INPUT"SATZNR";SA:IF SA<0 OR SA>63000 TH
   EN 60                                    <064>
64 IF I$="L"THEN PRINT"{CLR}";:GOTO 2000    <127>
66 IF I$="S"THEN PRINT"{CLR}";:MO=1:GOTO 3
   000                                      <089>
70 GOTO 50                                  <024>
2000 REM LESEN ---------------------------
     -------------------------------------
     -                                      <193>
2001 :PRINT"{CLR,RVSON}";SA;"{LEFT}-";STR$
     (SA+20);"{SPACE,RVOFF}"                <148>
2005 IA=SA                                  <247>
2007 FOR SA=IA TO IA+20                     <154>
2010 : GOSUB 9000:GOSUB 9100                <196>
2025 : IF A=50 THEN SA=IA+21:GOTO 2045      <162>
2030 : IF LEN(A$)<40 THEN PRINT A$          <211>
2035 : IF LEN(A$)>=40 THEN PRINT A$;        <118>
2045 NEXT:PRINT                             <127>
2046 IF A=50 THEN PRINT"SATZNUMMER ZU HOCH
     !"                                     <243>
2050 GOTO 50                                <228>
3000 REM SCHREIBEN -----------------------
     -------------------------------------
     -                                      <214>
3005 PRINT"EINGABESATZ{2SPACE}//=KEINE EIN
     GABE"                                  <236>
3010 GOSUB 9000    :REM POSITIONIEREN       <202>
```

**Listing 23. Der Relative-Dateien-Editor**

```
3020 IF A=50 THEN PRINT"SATZNUMMER ZU HOCH
     !":GOTO 50           :REM FEHLERKAN
     ALABFR.                               <130>
3030 GOSUB 50000:A$=BE$
                    :REM INPUT-ROUT.       <181>
3040 IF A$="//"THEN MO=0:GOTO 50:REM ENDE  <241>
3050 GOSUB 9200    :REM A$ SCHREIBEN       <083>
3060 SA=SA+1     :REM NAECHSTER SATZ       <225>
3070 GOTO 3010     :REM NEXT (LOOP)        <190>
6000 REM VERSETZEN -----------------------
     ------------------------------------
     -                                     <154>
6002 PRINT"{CLR,RVSON}VERSETZEN{RVOFF}"    <195>
6010 PRINT"{DOWN}BITTE GEBEN SIE ANFANG UN
     D ENDE DES BE- REICHES AN, DEN SIE "; <209>
6015 PRINT"VERSETZEN WOLLEN !"             <239>
6020 POKE 198,0:INPUT"{DOWN}ANFANG";AN     <211>
6021 POKE 198,0:INPUT"ENDE";EN:IF AN>EN TH
     EN PRINT"DAS IST UNSINN !":GOTO 6020  <113>
6023 SA=AN:GOSUB 9000:IF A=50 THEN PRINT"A
     NFANGSSATZ EXISTIERT NICHT!":GOTO 50  <229>
6024 SA=EN:GOSUB 9000:IF A=50 THEN PRINT"E
     NDSATZ ZU HOCH!":GOTO 50              <152>
6026 PRINT"{DOWN}QUELLDATEI = ";D$         <007>
6028 POKE 198,0:INPUT"ZIELDATEI ";Z$:IF Z$
     =""THEN Z$=D$:PRINT"{UP,13RIGHT}";D$  <098>
6029 PRINT"{DOWN}BITTE ANFANG DES NEUEN BE
     REICHS ANGEBEN!";:POKE 198,0:INPUT NB
     :PRINT                                <193>
6030 SA=NB:GOSUB 9000:IF A=50 THEN PRINT"N
     EUER ANFANGSSATZ ZU HOCH!":GOTO 50    <198>
6031 SA=NB+EN-AN:GOSUB 9000:IF A=50 THEN P
     RINT"BEREICH ZU GROSS!":GOTO 50       <240>
6032 PRINT"BITTE WARTEN!"                  <125>
6036 SA=AN:IM=0:IF AN=EN THEN PRINT"*** FE
     RTIG ***":POKE 198,0:WAIT 198,1:GOTO
     50                                    <113>
6038 GOSUB 9000:GOSUB 9100:IF A=50 THEN 61
     00                                    <203>
6040 M$(SA-AN)=A$                          <219>
6045 IF SA=EN THEN 6100                    <208>
6050 IF SA-AN=M%THEN IM=1:GOTO 6100:REM *  <108>
6060 SA=SA+1:GOTO 6038                     <094>
6100 REM TRANSPORT                         <068>
6110 PRINT"BITTE DISK MIT ZIELDATEI EINLEG
     EN !":POKE 198,0:WAIT 198,1           <204>
6120 CLOSE 1:PRINT#15,"I0":OPEN 1,8,3,Z$   <182>
6200 SA=NB:GOSUB 9000:IF A=50 THEN 50      <222>
6205 II=EN-AN:IF IM=1 THEN II=M%:REM *     <163>
6215 FOR I=0 TO II                         <014>
6220 A$=M$(I):IF A$=""THEN A$=":"          <108>
6225 GOSUB 9200:REM SCHREIBEN              <119>
6240 SA=SA+1:NEXT I:IF IM<>0 THEN 6260     <135>
6250 PRINT"BITTE DISK MIT QUELLDATEI EINLE
     GEN !":POKE 198,0:WAIT 198,1:GOTO 50  <185>
6260 AN=AN+M%:NB=NB+M%:PRINT"BITTE DISK MI
     T QUELLDATEI EINLEGEN !"              <248>
6270 POKE 198,0:WAIT 198,1:CLOSE 1:PRINT#1
     5,"I0":OPEN 1,8,3,D$:GOTO 6036:REM *  <096>
7000 REM PRINTER-ROUTINE -----------------
     ------------------------------------
     -                                     <065>
7001 PRINT"{CLR,RVSON}PRINTER-ROUTINE MPS
     801{RVOFF}                            <097>
7002 INPUT"{DOWN}ANFANG";AN:INPUT"ENDE";EN
     :IF AN>EN THEN 7000                   <176>
7005 SA=AN:GOSUB 9000:IF A=50 THEN PRINT"A
     NFANGSSATZ EXISTIERT NICHT!":GOTO 50  <195>
7006 SA=EN:GOSUB 9000:IF A=50 THEN PRINT"E
     NDSATZ ZU HOCH!":GOTO 50              <118>
7009 AN=AN-1:PRINT"{DOWN}BITTE WAEHLEN SIE
     :{DOWN}                               <025>
7010 PRINT,"NORMALE AUSGABE ..1            <248>
7011 PRINT,"D.Z.S. AUSGABE ...2            <165>
7012 PRINT,"EIGENE ROUTINE ...3            <072>
7013 GET I$:IF VAL(I$)<1 OR VAL(I$)>3 THEN
      7013                                 <176>
7014 IF I$="2"THEN 7100                    <044>
7015 IF I$="3"THEN 7500                    <205>
7020 REM NORMALE AUSGABE --------------    <016>
7021 OPEN 3,4,7:FOR SA=AN TO EN:GOSUB 9000
     :GOSUB 9100                           <049>
7025 IF A=50 THEN PRINT"SATZNUMMER ZU HOCH
     !":SA=EN+1:GOTO 7045                  <175>
7035 PRINT#3,RIGHT$("{3SPACE}"+STR$(SA),3)
     ;"{4SPACE}";                          <085>
7036 FOR I=1 TO LEN(A$)                    <224>
7037 :IF MID$(A$,I,1)="{RIGHT}"THEN PRINT#
     3," ";:GOTO 7040                      <137>
7038 :PRINT#3,MID$(A$,I,1);                <048>
7040 NEXT I:PRINT#3                        <140>
7045 NEXT SA:CLOSE 3:PRINT:GOTO 50         <117>
7100 REM D.Z.SATZ AUSGABE -------------    <230>
7101 OPEN 3,4,7:FOR SA=AN TO EN:GOSUB 9000
     :GOSUB 9100                           <129>
7102 IF A=50 THEN SA=EN+1:PRINT"SATZNUMMER
      ZU HOCH!":GOTO 7170                  <022>
7110 PRINT#3,RIGHT$("{4SPACE}"+STR$(SA),4)
     ;"{4SPACE}";                          <161>
7120 FOR I=1 TO LEN(A$):IF A$=""THEN 7160  <163>
7125 IM=ASC(MID$(A$,I,1))                  <201>
7126 IF IM>64 AND IM<91 OR IM>96 AND IM<12
     3 THEN 7160                           <065>
7130 IF IM=64 THEN PRINT#3,CHR$(34);:GOTO
     7170                                  <022>
7135 IF IM=173 THEN PRINT#3,"BE";:GOTO 717
     0                                     <125>
7140 IF IM=174 THEN PRINT#3,"SS";:GOTO 717
     0                                     <238>
7145 IF IM=176 THEN PRINT#3,"AE";:GOTO 717
     0                                     <106>
7150 IF IM=181 THEN PRINT#3,"UE";:GOTO 717
     0                                     <016>
7152 IF IM=182 THEN PRINT#3,"OE";:GOTO 717
     0                                     <145>
7154 IF IM=184 THEN PRINT#3,"UE";:GOTO 717
     0                                     <247>
7156 IF IM=185 THEN PRINT#3,"OE";:GOTO 717
     0                                     <120>
7158 IF IM=94 THEN PRINT#3,":";  :GOTO 717
     0                                     <244>
7159 IF IM=95 THEN PRINT#3,",";  :GOTO 717
     0                                     <179>
7160 PRINT#3,CHR$(IM);                     <028>
7170 NEXT I:PRINT#3,CHR$(13);              <147>
7175 NEXT SA:CLOSE 3:PRINT:GOTO 50         <249>
7500 REM EIGENE ROUTINE --------------     <183>
7510 PRINT"{DOWN}SIE HABEN NOCH KEINE EIGE
     NE PRINTER-{4SPACE}ROUTINE DEFFINIERT
     !":GOTO 50                            <032>
8000 REM TEXT ZENTRIEREN -----------------
     ------------------------------------
     -                                     <058>
8001 I$="":FOR I=1 TO 10:I$=I$+"{20RIGHT}"
     :NEXT I                               <067>
8002 REM SL=27                             <234>
8005 REM DIMM$(300):GOTO8100               <009>
8010 PRINT"{RVSON}ZENTRIEREN{RVOFF,DOWN}"  <040>
8015 PRINT"BITTE GEBEN SIE ANFANG UND ENDE
      DES{5SPACE}BEREICHES AN, DER ZENTRIE
     RT";                                  <197>
8017 PRINT" WERDEN SOLL!"                  <039>
8020 INPUT"ANFANG";AN:INPUT"ENDE";EN:IF AN
     >EN THEN 50                           <157>
8025 SA=AN:GOSUB 9000:IF A=50 THEN PRINT"A
     NFANGSSATZ EXISTIERT NICHT!":GOTO 50  <199>
8030 SA=EN:GOSUB 9000:IF A=50 THEN PRINT"E
     NDSATZ ZU HOCH!":GOTO 50              <126>
8040 PRINT"{DOWN}LINKS-RECHTS BUENDIG ZENT
     RIEREN{2SPACE}- '1'                   <098>
8045 PRINT"{DOWN}IN DIE MITTE ZENTRIEREN{1
     0SPACE}- '2'                          <045>
8050 GET II$:IF II$<>"1"AND II$<>"2"THEN 8
     050                                   <115>
8060 IF II$="2"THEN 8600                   <081>
8099 REM LINKS-RECHTS BUENDIG ----------   <089>
8100 PRINT"{DOWN}BITTE WARTEN!"            <255>
8105 FOR SA=AN TO EN                       <207>
8110 : GOSUB 9000:GOSUB 9100:PRINT A$:PRIN
     T LEN(A$)                             <043>
8120 : IF RIGHT$(A$,1)=CHR$(29)THEN A$=LEF
     T$(A$,LEN(A$)-1):GOTO 8120            <192>
8122 : IF RIGHT$(A$,1)=CHR$(32)THEN A$=LEF
     T$(A$,LEN(A$)-1):GOTO 8122            <128>
8123 : IF RIGHT$(A$,1)=CHR$(160)THEN A$=LE
     FT$(A$,LEN(A$)-1):GOTO 8123           <078>
8130 : IF LEN(A$)=SL THEN 8520             <059>
8200 :  LU=SL-LEN(A$):PRINT"LU=";LU        <233>
8220 :  I=1:IC=0:REM BEGIN LOOP -------    <026>
8222 :                                     <146>
8230 :   IF MID$(A$,I,1)=""OR MID$(A$,I+1,
     1)=""THEN 8530                        <222>
```

```
8231 :    IM=ASC(MID$(A$,I,1))                <089>
8232 :    IN=ASC(MID$(A$,I+1,1))              <178>
8240 :    IF IM<>29 AND IM<>32 AND IM<>160
     THEN 8500                                <208>
8250 :    IF IN=29 OR IN=32 OR IN=160 THEN
     8500                                     <056>
8260 :     A$=LEFT$(A$,I)+" "+RIGHT$(A$,LEN
     (A$)-I)                                  <164>
8261 :    IF MID$(A$,I,1)=" "OR MID$(A$,I,1
     )="{RIGHT}"THEN I=I+1:GOTO 8261          <058>
8265 :    LU=LU-1:IF LU=0 THEN 8510           <028>
8266 :    IC=IC+1                             <167>
8270 :    I=I+1                               <177>
8400 :                                        <068>
8500 I=I+1                                    <101>
8505 IF I<LEN(A$)THEN 8230                    <186>
8510 IF I=LEN(A$)AND LU>0 AND IC<>0 THEN 8
     220                                      <035>
8520 GOSUB 9000:GOSUB 9200                    <006>
8530 NEXT SA                                  <076>
8540 GOTO 50                                  <112>
8600 REM IN DIE MITTE -----------------       <133>
8610 PRINT"{DOWN}BITTE WARTEN!"               <001>
8620 FOR SA=AN TO EN                          <214>
8630 :GOSUB 9000:GOSUB 9100                   <212>
8635 IF RIGHT$(A$,1)=CHR$(29)THEN A$=LEFT$
     (A$,LEN(A$)-1):GOTO 8635                 <152>
8636 IF RIGHT$(A$,1)=CHR$(32)THEN A$=LEFT$
     (A$,LEN(A$)-1):GOTO 8636                 <055>
8637 IF RIGHT$(A$,1)=CHR$(160)THEN A$=LEFT
     $(A$,LEN(A$)-1):GOTO 8637                <146>
8640 :IF LEN(A$)=SL THEN 8680                 <128>
8650 :AS=SL-LEN(A$): REM ANZAHL SPACES        <234>
8660 :A$=LEFT$(I$,AS/2)+A$                    <247>
8670 :GOSUB 9000:GOSUB 9200                   <004>
8680 NEXT SA                                  <226>
8690 GOTO 50                                  <006>
9000 REM POSITIONIEREN ****************       <130>
9010 HB%=SA/256:LB%=SA-HB%*256                <068>
9020 PRINT#15,"P";CHR$(3);CHR$(LB%);CHR$(H
     B%);CHR$(1)                              <005>
9030 INPUT#15,A                               <079>
9040 RETURN                                   <208>
9100 REM SATZ EINLESEN ****************       <042>
9110 IF A=50 THEN RETURN                      <082>
9120 INPUT#1,A$                               <116>
9140 RETURN                                   <052>
9200 REM SATZ SCHREIBEN ***************       <106>
9210 IF A=50 THEN RETURN                      <182>
9211 IF A$=""THEN A$=":":REM ZUR SICHERHEI
     T !!!                                    <205>
9220 PRINT#1,A$                               <042>
9240 RETURN                                   <154>
9900 REM BEENDEN **********************       <083>
9910 CLOSE 1                                  <013>
9920 CLOSE 15                                 <106>
9930 PRINT "++ ENDE ++"                       <017>
9940 END                                      <034>
10000 REM SORT-ALOGARITHMUS --------------
      ----------------------------------------
      --                                      <090>
10010 PRINT"{CLR,RVSON}DATEI SORTIEREN{RVO
      FF}"                                    <124>
10100 PRINT"BITTE GEBEN SIE ANFANG UND END
      E DES ZU{2SPACE}SORTIERENDEN BEREICH
      S AN !                                  <096>
10110 INPUT"ANFANG";AN                        <027>
10120 INPUT"ENDE";   EN                       <236>
10122 SA=AN:GOSUB 9000:IF A=50 THEN PRINT"
      ANFANGSSATZ EXISTIERT NICHT!":GOTO 5
      0                                       <008>
10124 SA=EN:GOSUB 9000:IF A=50 THEN PRINT"
      ENDSATZ ZU HOCH!":GOTO 50               <188>
10130 IF EN-AN>M%THEN PRINT"BEREICH ZU GRO
      SS!":GOTO 50                            <041>
10135 PRINT"{DOWN}BITTE MARKIEREN SIE DEN
      BEREICH INNER-                          <106>
10136 PRINT"HALB DER STRINGS NACH DEM DIE
      SORTIERUNG";                            <205>
10137 PRINT"ERFOLGEN SOLL!"                   <081>
10138 SA=AN:GOSUB 9000:GOSUB 9100:IF A=50
      THEN PRINT"SATZ EXISTIERT NICHT!":GO
      TO 50                                   <124>
10139 IG=-1:GOSUB 60000:IG=0                  <050>
10140 IC=0:FOR SA=AN TO EN:IC=IC+1            <237>
10145 :GOSUB 9000:GOSUB 9100:IF A=50 THEN
      PRINT"SATZNUMMER ZU HOCH!":SA=EN+1:G
      OTO 1015                                <187>
10146 :M$(IC)=A$                              <048>
10150 NEXT SA                                 <172>
10180 Z=0:LG(1)=1:RG(1)=EN-AN+1               <053>
10190 GOSUB 10200:GOTO 10400                  <047>
10200 REM  QUICKSORT ----------------         <135>
10210 Z=Z+1:IF LG(Z)>=RG(Z)THEN 10350         <058>
10220 X=LG(Z):Y=RG(Z)                         <023>
10230 VG$=MID$(M$(INT((X+Y)/2)),MA(1),MB(1
      )-MA(1)+1)                              <084>
10240 IF X>Y THEN 10330                       <234>
10250 IF MID$(M$(X),MA(1),MB(1)-MA(1)+1)<V
      G$THEN X=X+1:GOTO 10250                 <069>
10260 IF MID$(M$(Y),MA(1),MB(1)-MA(1)+1)>V
      G$THEN Y=Y-1:GOTO 10260                 <095>
10270 IF X>Y THEN 10330                       <008>
10280 S$=M$(X):M$(X)=M$(Y):M$(Y)=S$           <143>
10290 X=X+1:Y=Y-1:GOTO 10240                  <116>
10330 RG(Z+1)=Y:LG(Z+1)=LG(Z):GOSUB 10210     <020>
10340 LG(Z+1)=X:RG(Z+1)=RG(Z):GOSUB 10210     <196>
10350 Z=Z-1:RETURN                            <205>
10400 REM ABSPEICHERN                         <132>
10410 IC=0:FOR SA=AN TO EN:IC=IC+1            <253>
10420 :GOSUB 9000:A$=M$(IC):GOSUB 9200        <226>
10430 NEXT SA                                 <198>
10500 GOTO 50                                 <040>
20000 REM KORREKTUR-ROUTINE --------------
      ----------------------------------------
      --                                      <195>
20001 FOR I=0 TO LEN(BE$):PRINT CHR$(20);:
      NEXT I:BE$=""                           <010>
20010 SA=SA+SC        : REM TEXTZEIGER        <162>
20015 IF SA<0 THEN SA=0 : REM SPERRE          <008>
20020 GOSUB 9000      : REM POSITIONIEREN     <194>
20025 IF A=50 THEN SA=SA-SC:GOTO 20020        <223>
20040 INPUT#1,A$                              <114>
20060 RETURN                                  <050>
30000 REM DIRECTORY EINLESEN -------------
      ----------------------------------------
      --                                      <161>
30010 PRINT"{CLR,RVSON}INHALTSVERZEICHNIS:
      {DOWN,RVOFF}"                           <199>
30110 PRINT#15,"IO":OPEN 2,8,2,"#"            <007>
30120 T=18:S=1                                <221>
30130 PRINT#15,"U1";2;0;T;S                   <021>
30140 PRINT#15,"B-P";2;0                      <214>
30150 GET#2,X$:IF X$=""THEN X$=CHR$(0)        <054>
30160 T=ASC(X$)                               <098>
30170 GET#2,X$:IF X$=""THEN X$=CHR$(0)        <074>
30180 S=ASC(X$)                               <114>
30190 FOR X=0 TO 7                            <149>
30192 PRINT#15,"B-P";2;X*32+2                 <060>
30194 GET#2,X$:IF X$=""THEN X$=CHR$(0)        <098>
30196 IF ASC(X$)=0 THEN 30290                 <100>
30200 PRINT#15,"B-P";2;X*32+5                 <092>
30210 FF$=""                                  <106>
30220 FOR Y=0 TO 15                           <239>
30230 GET#2,X$:IF X$=""THEN X$=CHR$(0)        <136>
30240 IF ASC(X$)=160 THEN 30270               <191>
30250 FF$=FF$+X$                              <168>
30260 NEXT Y                                  <244>
30270 IF A=0 THEN A=1:PRINT FF$;:GOTO 3029
      0                                       <166>
30280 A=0:PRINT TAB(19);"[illegible] ";FF$    <012>
30290 NEXT X                                  <010>
30300 IF T<>0 THEN 30130                      <091>
30310 CLOSE 2:PRINT                           <059>
30330 GOTO 50                                 <056>
50000 REM TEXTEINGABE --------------------
      ----------------------------------------
      --                                      <106>
50002 IF AR=1 THEN AR=0:A$="{RVSON}SATZNUM
      MER ZU HOCH!{RVOFF}":GOTO 50007         <097>
50006 GOSUB 9000:GOSUB 9100                   <085>
50007 PRINT"{HOME,40SPACE}";:PRINT"{HOME,R
      VSON}SATZ";SA;"{RVOFF}"                 <168>
50008 PRINT"{40SPACE,UP}";A$                  <069>
50009 PRINT"TTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
      TTTTTTTTTT";                            <204>
50010 PRINT"{DOWN}"                           <173>
50020 BE$="":POKE 198,0:POKE 211,0:POKE 21
      4,22:SYS 58732:PRINT"{YELLOW}[illegible]";  <045>
```

**Listing 23. Der Relative-Dateien-Editor (Fortsetzung)**

# 64'er HARDWARE-SERVICE

Bestellungen aus anderen Ländern bitte per Auslandspostanweisung!

Bestellungen aus der Schweiz bitte direkt an:
Markt&Technik Vertriebs AG
Kollerstrasse 3
CH-6300 Zug
Tel. 042/41 56 56

Bestellungen aus Österreich bitte direkt an:
Ueberreuter Media
Handels- und Verlagsges. mbH,
Alser Straße 24,
1091 Wien
Tel. 0222/48 15 38-0

## Hardware für alle – ein neuer 64'er Leser-Service

Der Commodore 64 hat schon oft bewiesen, wie vielseitig er ist. Er läßt sich nicht nur mit Programmen, sondern auch durch so manche Hardware-Erweiterung sinnvoll nutzen und ausbauen. Dabei ist es sicherlich ein reizvoller Bestandteil des Computer-Hobbys, sich solche Erweiterungen selbst nachzubauen. Aber nicht jeder Leser verfügt über die Gelegenheit und Zeit zur Platinenherstellung. Hinzu kommt, daß es oft zu teuer ist, wegen einer bestimmten Erweiterung Investitionen von mehreren hundert Mark für eine Platinenstation zu tätigen. Die in der 64'er abgedruckten Hardware-Erweiterungen sind in drei verschiedenen Ausbaustufen zu erhalten:

**1. Als Platinen**
Nur Leerplatinen. Die Beschaffung der Bauteile und der Zusammenbau bleiben bei Ihnen.

**2. Als Bausätze**
Unsere Bausätze enthalten alle Teile, die notwendig sind, um die beschriebene Erweiterung komplett aufzubauen. Sie brauchen die Bauteile nur noch, gemäß der Anleitung im Heft, zusammenzulöten und einzubauen.

**3. Als Fertiggeräte**
Die Fertiggeräte sind komplett aufgebaute und geprüfte Geräte. Sie brauchen die Erweiterung lediglich noch einzubauen.

### Qualität & Service

- Die 64'er Hardware hat einen hohen Qualitätsstandard. Wir verwenden nur beste Epoxid-Harz-Platinen mit Lötstopp-Lack.
- Wir verwenden nur Präzisionssockel mit gedrehten Kontakten.
- Alle Platinen werden professionell gefertigt. Wenn notwendig mit doppelseitiger Beschichtung und Löt-Durchkontaktierungen.
- Jedes Gerät, das wir versenden, wurde auf Funktionstüchtigkeit geprüft.
- Wir sind auch nach dem Verkauf für Sie da. Neben der gesetzlichen Garantie bietet unser Service- und Fertigungspartner Ihnen Hilfe und Unterstützung an.

### Einbauservice

Für die Angebote 4 (Super Kernal) und 5 (64'er DOS) bieten wir einen Einbauservice an. Jeder Lieferung dieser Produkte liegt neben der detaillierten Einbauanleitung ein Angebot zum kostengünstigen Umbau Ihres C64 beziehungsweise Ihrer 1541 Floppy bei. Falls Ihr C64 keine gesockelten Bausteine besitzt, können Sie dort ebenfalls hochwertige Stecksockel einbauen lassen.

### Unsere Garantie

Im Rahmen der Versand- und Lieferbedingungen unterliegen die Geräte einer Gewährleistungszeit von 6 Monaten ab Lieferung. Der Lieferung liegt eine Service-Karte bei, die Sie im Falle einer Beanstandung zusammen mit dem Gerät an die auf der Karte vermerkte Adresse schicken können. Die gleiche Karte verwenden Sie bitte bei Reparaturen nach der Garantiezeit.

## Unser Angebot

**Angebot 1: Expansion-Port EPROM-Platine mit 1 x 8 KByte Speicherplatz für 2732 bis 2764 EPROMS.**
Beschreibung in Ausgabe 10/85
Bestellnummer: HW 010
**DM 19,80*** (sFr. 17,50)
Dieser Artikel wird nur als Fertiggerät angeboten.

**Angebot 2: Expansion-Port EPROM-Platine mit 2 x 8 KByte Speicherplatz für 2732 bis 2764 EPROMS, mit Umschaltmöglichkeit**
Beschreibung in Ausgabe 10/85
Leerplatine
Bestellnummer: HW 020
**DM 24,80*** (sFr. 22,-)
Bausatz mit allen Teilen:
Bestellnummer: HW 021
**DM 49,80*** (sFr. 43,-)
Fertiggerät, getestet, wie beschrieben:
Bestellnummer: HW 022
**DM 59,80*** (sFr. 51,-)

**Angebot 3: EPROM Trans – Die Speichererweiterung**
ROM-Speichererweiterung zum Einbau in den C 64, gleichzeitig Steckplatz für ein Original- oder ein alternatives Betriebssystem. Zwei Platinen in Epoxid-Harz-Ausführung wie in Ausgabe 10/85 beschrieben.
Leerplatine
Bestellnummer: HW 030
**DM 49,80*** (sFr. 43,-)
Bausatz mit allen Teilen:
Bestellnummer: HW 031
**DM 119,80*** (sFr. 102,-)
EPROM Trans ist nicht als Fertiggerät erhältlich.

**Angebot 4: Super Kernal**
Erweitertes Betriebssystem für den C64 mit vielen neuen Funktionen, inkl. Adaptersockel, einbaufertig in den C64.
Beschreibung in Ausgabe 11/85
Version 1: Enthält Hypra Load / DOS 5.1 / Funktionstastenbelegung / Renew / RS232
Bestellnummer: HW 040
Version 2: Enthält Hypra Load / DOS 5.1 / Funktionstastenbelegung / Renew / Super Centronics Schnittstelle
Bestellnummer: HW 041
Version 3: Enthält Hypra Load / DOS 5.1 / Funktionstastenbelegung / Renew / Hypra Save
Bestellnummer: HW 042
Version 4: Enthält Hypra Load / DOS 5.1 / Funktionstasten / Hypra Save / Centronics klein
Bestellnummer: HW 043
Jede Version kostet:
**DM 39,80*** (sFr. 34,-)

* Preise inkl. Mehrwertsteuer

**Angebot 5: 64'er DOS**
Jetzt wird das 1541 Laufwerk zum Renner. Mit wenig Aufwand beschleunigt 64'er DOS alle Funktionen des Laufwerkes. Das neue Betriebssystem für den Commodore 64 und das 1541 Laufwerk ist auf 2 Speicher-EPROMs der Sorte 2764 untergebracht und inkl. Adaptersockel einbaufertig vorbereitet.
Beschreibung in Ausgabe 3/86 (Einbauanleitung liegt bei).
Preis für beide EPROMs inkl. Adaptersockel
Bestellnummer: HW 050
**DM 69,-*** (sFr. 59,-)
Lieferbar ab Februar/März 1986

**Angebot 6: EPROM-Programmiergerät**
Programmiergerät für EPROMs der Typen: 2532, 2732, 2764, 27128, 27256, 27512. Platine aus Epoxid-Harz für Expansion-Port.
Beschreibung in den Ausgaben 12/85, 1/86 und 2/86.
Leerplatine
Bestellnummer: HW 060
**DM 44,80*** (sFr. 39,90)
Lieferbar ab März/April 1986
Spannungswandler
Bestellnummer: HW 061
**DM 19,80*** (sFr. 17,50)
Lieferbar ab März/April 1986
Betriebssoftware auf Diskette
Bestellnummer: HW 062
**DM 14,80*** (sFr. 13,90)
Lieferbar ab März/April 1986
Die Betriebssoftware befindet sich außerdem auf der Leserservice-Diskette der Ausgabe 2/86.
Kombinationsangebot
Leerplatine, Spannungswandler und Diskette im Paket.
Bestellnummer: HW 063
**DM 69,80*** (sFr. 59,50)
Lieferbar ab März/April 1986

**Angebot 7: Akustikkoppler HITRANS 300 C**
Mit einem Akustikkoppler öffnen Sie Ihrem Computer das Tor zur ganzen Welt. Der HITRANS 300 C stach im Akustikkoppler-Test der Ausgabe 3/86 durch die besten Übertragungseigenschaften hervor. Sie erhalten ihn bei uns als Fertiggerät, lediglich eine Blockbatterie muß eingesetzt und das Gehäuse zugeschraubt werden. Sie können den Koppler auch über ein 12-Volt-Netzteil, das in jedem Elektronikgeschäft preisgünstig erhältlich ist, betreiben. Die Bauanleitung für ein RS232-Interface finden Sie in der Ausgabe 3/85.
Preis für Akustikkoppler HITRANS 300 C (ohne Batterie)
**DM 248,-*** (sFr. 225,-)
Bestellnummer: HW 070
Betriebssoftware auf Diskette
Bestellnummer: HW 071
**DM 14,80*** (sFr. 13,90)
Die Betriebssoftware befindet sich außerdem auf der Programm-Service-Diskette des 64er-Sonderheftes SH7/85.

Bitte verwenden Sie für Ihre Bestellung immer die abgedruckte Postgiro-Zahlkarte oder einen Verrechnungsscheck. Sie erleichtern uns damit die Auftragsabwicklung, und dafür berechnen wir Ihnen keine Versandkosten.

```
50030 GET X$:IF PEEK(203)=1 THEN 50120        <226>
50035 IF PEEK(203)=7 AND PEEK(653)=0 THEN
      SC=1 : GOSUB 20000 : GOTO 50000         <023>
50037 IF PEEK(203)=7 AND PEEK(653)=1 THEN
      SC=-1 : GOSUB 20000 : GOTO 50000        <031>
50040 IF X$=""THEN 50030                      <171>
50050 IF LEN(BE$)=0 AND ASC(X$)=20 THEN 50
      030                                     <148>
50060 I=ASC(X$):IF I<32 OR I>133 AND I<159
       THEN IF I<>20 THEN 50030               <131>
50070 IF LEN(BE$)=SL AND I<>20 THEN 50030     <128>
50075 IF X$=" "THEN X$="{RIGHT}"              <012>
50080 BE$=BE$+X$                              <031>
50090 PRINT CHR$(20);X$;"[illegible]";        <108>
50100 IF I=20 THEN BE$=LEFT$(BE$,LEN(BE$)-
      2):GOTO 50030                           <101>
50110 GOTO 50030                              <001>
50120 PRINT"{LEFT,SPACE}":GOSUB 9000          <017>
50130 IF BE$=""THEN BE$=":"                   <049>
50140 PRINT"{HOME,24DOWN}":RETURN             <030>
60000 REM MARKIEREN WIE DATIERT WIRD -----
      -----------------------------------
      --                                      <071>
60010 I$=A$:FOR I=1 TO LEN(I$):M$(I)="0":N
      EXT                                     <131>
60020 IM=1:PRINT                              <131>
60025 PRINT I$:FOR I=1 TO LEN(I$):PRINT M$
      (I);:NEXT:PRINT                         <223>
60030 PRINT"{DOWN}EDIT{3SPACE}>";MID$(I$,I
      M,1);"<{4SPACE,RVSON}"IM"{RVOFF}"       <232>
60035 I=PEEK(203):IF I=64 THEN 60035          <110>
60040 IF I=2 AND PEEK(653)=0 AND IM<>LEN(I
      $)THEN IM=IM+1:GOTO 60070               <174>
60042 IF I=2 AND PEEK(653)=1 AND IM<>1 THE
      N IM=IM-1:GOTO 60070                    <119>
60045 IF I=56 THEN M$(IM)="1":GOTO 60070      <041>
60055 IF I=35 THEN M$(IM)="0":GOTO 60070      <178>
60056 IF I=1 THEN 60100                       <013>
60060 GOTO 60035                              <183>
60070 PRINT"{UP,25SPACE,25LEFT,3UP}";:GOTO
       60025                                  <139>
60100 REM UMRECHNEN                           <252>
60102 FOR I=1 TO 3:MA(I)=0:MB(I)=0:NEXT I     <148>
60105 FOR II=1 TO 3                           <169>
60110 :FOR I=1 TO LEN(I$):IF MA(II)=0 AND
      M$(I)=RIGHT$(STR$(II),1)THEN MA(II)=
      I                                       <059>
60113 NEXT I                                  <250>
60115 :IF MA(II)=0 THEN 60150                 <245>
60120 :   MB(II)=MA(II)                       <124>
60125 :   MB(II)=MB(II)+1:REM PRINT,MB(II)    <069>
60130 :   IF M$(MB(II))<>RIGHT$(STR$(II),1
      )THEN MB(II)=MB(II)-1:GOTO 60150        <217>
60135 :   IF MB(II)=LEN(I$)THEN 60150         <074>
60140 :   GOTO 60125                          <248>
60150 NEXT II                                 <179>
60170 PRINT"DIE SORTIERUNG ERFOLGT NACH FO
      LGENDEM{3SPACE}STRINGBEREICH:"          <088>
60180 PRINT"{RVSON}"MID$(I$,MA(1),MB(1)-MA
      (1)+1)"{RVOFF}"                         <168>
60190 RETURN                                  <047>
61000 REM FLOPPY-FEHLERKANAL LESEN -------
      -----------------------------------
      --                                      <254>
61005 PRINT"{CLR,RVSON}FLOPPY-FEHLERKANALA
      BFRAGE{RVOFF}"                          <208>
61010 INPUT#15,A,B$,C,D                       <166>
61020 PRINT A,B$,C,D                          <126>
61040 GOTO 50                                 <031>
```

**Listing 23. Der Relative-Dateien-Editor (Schluß)**

**Dokumentation zum Listing 23**

Wichtigste Variablen:

| | |
|---|---|
| M% | Größe des Stringpuffers |
| M$(1)-M$(M%), M(1)-M(M$) | Stringpuffer |
| LG, RG | werden für Quicksort-Unterprogramme benötigt |
| A$ | von relativer Datei eingelesener String beziehungsweise String, der auf relative Datei geschrieben wird. |
| SA | Satz, auf den der Zeiger der relativen Datei positioniert ist beziehungsweise werden soll. |
| D$ | Filename der Datei |
| LB, HB | Lowbyte, Highbyte |
| AN, EN | Anfang beziehungsweise Ende eines Bereiches der relativen Datei, der sortiert beziehungsweise ausgedruckt werden soll. |



Variablen beziehungsweise Stringnamen, die mit I beginnen, sind Hilfsvariablen beziehungsweise Strings, die sich während dem Programmablauf ständig ändern – zum Beispiel Schleifen, Prüfvariablen etc.

| | |
|---|---|
| **0 - 15** | Vorbereitungen, Felder dimensionieren etc. |
| **19 - 30** | Anfangsmenü |
| **31 - 41** | Anlegen einer neuen Datei. |
| **50 - 70** | Zwischenmenü beziehungsweise Menüzeile. Dies ist der Hauptteil des Programmes. Von hier aus werden alle möglichen Routinen angewählt. |
| **2000 - 2050** | Routine zum Lesen von Sätzen einer relativen Datei. |
| **3000 - 3070** | Routine zum Schreiben von Sätzen in eine relative Datei. |
| **6000 - 6270** | Routine zum Versetzen bestimmter Bereiche einer relativen Datei innerhalb derselben oder auf andere relative Dateien. Die Daten werden jeweils zuerst von der Quelldatei in den Puffer M% gelesen. Dann wird die Quelldatei geschlossen und die Zieldatei geöffnet. Sodann wird der Pufferinhalt in die Zieldatei übertragen etc. |
| **7000 - 7510** | Printer-Routine zum Ausdrucken von relativen Dateien. |
| **7500 - 7999** | In diesem Bereich können Sie eine eigene Printer-Routine unterbringen. So kann zum Beispiel der deutsche Zeichensatz Ihrem Drucker angepaßt werden. |
| **8000 - 8690** | Verschiedene Routinen zur formatierten Textausgabe einer relativen Datei. |
| **9000 - 9140** | Die bereits vorgestellten Unterprogramme zum Lesen, Schreiben, Positionieren und Schließen einer Datei. |
| **10000 - 10500** | Routine zum Sortieren eines bestimmten Bereiches innerhalb der Datei. Der gewünschte Bereich, der aus maximal M% Sätzen bestehen darf, wird in den Stringpuffer eingelesen. Nun wird der Stringpuffer mittels einem Quicksort-Algorithmus (10200 - 10350) sortiert und anschließend wieder in die Datei auf Disk zurückgeschrieben. |
| **20000 - 20060** | Korrekturroutine – »Blättern« in einer relativen Datei mittels den Cursor-Tasten. Diese Routine wird von der Texteingaberoutine ab 50000 aufgerufen. |
| **30000 - 30330** | Unterprogramm zum Einlesen des Disk-Directories ohne Programmverlust. |
| **50000 - 50140** | Modifizierte INPUT-Routine. |
| **60000 - 60190** | Unterprogramm zum Markieren des Stringbereiches, der beim Sortieren als Kriterium gelten soll. |
| **61000 - 61040** | Unterprogramm zur Abfrage des Fehlerkanals des Floppy-Laufwerks. |

Die Dokumentation ist sehr oberflächlich; ich glaube nicht, daß am Editor noch viel verbessert werden kann. Mit dem Edi-

tor bekommen Sie relative Dateien in den Griff. Mit Hilfe der drei Unterprogramme zum Lesen, Schreiben und Positionieren können Sie sich leicht die unterschiedlichsten Dateiverwaltungen mittels kurzer Basic-Programme zusammenbasteln. Dazu müssen Sie jedoch wissen, wie mit dem Editor umzugehen ist:

**Bedienungsanleitung zum Texteditor**

Nach dem Programmstart mittels RUN erscheint das Grundmenü. Zum einen kann von hier aus ein alternativer Zeichensatz geladen werden, der mit dem Zeichensatz-Editor erstellt werden beziehungsweise ab hexadezimal E000 liegen muß. Der Zeichensatz muß den Namen »D.Z.SATZ« haben (beziehungsweise Name im Listing Zeile 27 umändern). Es bietet sich also an, mit dem Zeichensatz-Editor einen deutschen Zeichensatz automatisch generieren zu lassen und ihn dann als D.Z.SATZ auf die Disk, auf der sich auch der Editor befindet, zu speichern. Zum Laden des Alternativ-Zeichensatzes muß einfach die Taste »D« gedrückt werden. Wenn das Disketten-Laufwerk wieder zum Stillstand kommt, befindet sich der neue Zeichensatz im Computer. Natürlich hätte ich nach dem Filenamen des Zeichensatzes fragen können, nachdem »D« gedrückt wurde.

Ich habe mich jedoch für diese Lösung entschieden, da sich dadurch viel schneller mit dem Texteditor arbeiten läßt: Man muß so nicht erst mehrere Parameter eingeben, bevor man mit dem Arbeiten beginnen kann.

Mittels »N« kann eine neue Datei angelegt werden. Sie müssen lediglich den Dateinamen (Filenamen), maximal erlaubte Satzlänge und die Satzanzahl der Datei angeben. Dann eine Disk mit genügend freiem Platz einlegen und RETURN drücken. Je nach Größe der Datei kann es einige Zeit in Anspruch nehmen, bis der Computer hiermit fertig ist. Das Anlegen einer relativen Datei wird so zum Kinderspiel. Mittels »A« wird eine alte Datei zum weiteren Bearbeiten geöffnet. Es ist hierzu erforderlich, daß sich die Disk, auf der sich die Datei befindet, im Laufwerk befindet (logisch). Ganz Eilige können auch einfach mehrmals die RETURN-Taste drücken. Dann wird einfach das erste Programm der Disk, die sich im Laufwerk befindet, geladen und die maximale Zeichenanzahl auf 40 festgelegt.

Achtung: Beim Anlegen einer relativen Datei ist immer folgendes zu beachten: Soll die Satzlänge einer Datei 40 Zeichen betragen, so muß man beim Anlegen der Datei als maximale Satzlänge 41 angeben – also immer eins mehr als die Satzlänge später tatsächlich sein soll. Dieser eine Platz beziehungsweise Zeichen wird nämlich für das CHR$(13) (Carriage Return) benötigt, das am Satzende jeweils angehängt wird! Es ist in bedingten Fällen auch möglich, auf das CHR$(13) zu verzichten. Man muß dann jedoch »ganz cool« über die zahlreich auftretenden FILE DATA beziehungsweise OVERFLOW IN RECORD ERRORs hinwegsehen!

Ich rate Ihnen jedenfalls, nicht an diesem einen Zeichen zu sparen!

Nach Anwahl des ersten oder zweiten Grundmenü-Punktes (alte Datei eröffnen beziehungsweise neue anlegen) gelangt man ins Hauptmenü, das aus einer einzigen Zeile besteht:

```
s/l/v/o/d/i/z/f/ende
```

Die einzelnen Buchstaben beziehungsweise Tasten stehen für folgende Funktionen:

s – schreiben
l – lesen
v – versetzen
o – ordnen
d – drucken
i – Inhaltsverzeichnis
z – zentrieren
f – Fehlerkanal abfragen
ende – na, was wohl?

## Die Menüpunkte im einzelnen

**Schreiben**

Zuerst muß die Satznummer eingegeben werden, ab der man schreiben will. Danach erscheint das Satzeingabe-Bild. In der obersten Bildschirmzeile sind Satznummer sowie momentaner Inhalt des Satzes zu sehen. Der Cursor steht in der untersten Zeile des Bildschirms. Dort kann ein neuer Satzinhalt eingegeben werden. Die Cursor-Tasten sind hierbei in ihrer Funktion zum Bewegen des eigentlichen Cursors außer Gefecht gesetzt. Auch alle anderen Steuertasten (zum Beispiel Farbwahl und CLR), außer der DEL-Funktion, funktionieren nicht mehr. Wie ich anfangs bereits sagte, kann bei einem Texteditor für relative Dateien keineswegs an die Vorzüge eines Textverarbeitungssystems gedacht werden. Sie können lediglich gesamte Sätze schreiben und diese mit RETURN absenden, wobei Sie automatisch in die nächste Zeile gelangen. Mit den Tasten CURSOR UP und CURSOR DOWN kann der Text nach oben beziehungsweise unten gescrollt werden.

Durch Eingabe von // wird die Texteingabe beendet – das Hauptmenü erscheint wieder.

Einen gewaltigen Vorteil hat ein Relative-Dateien-Editor allerdings gegenüber einem Textverarbeitungssystem: Bei einem Stromausfall gehen nur die Daten verloren, die sich gerade im Puffer des Floppy-Laufwerks befinden (meist nur 2 – 3 Sätze bei 40 Zeichen Satzlänge)!

**Lesen**

Auch hier muß nur die Satznummer des Satzes eingegeben werden, ab der gelesen werden soll. Auf dem Bildschirm erscheinen sodann die 20 Sätze ab dem eingegebenen Wert. Wenn Sie nach der Satznummer gefragt werden und einfach die RETURN-Taste drücken, ohne eine Zahl einzugeben, so wird ab dort weitergelesen beziehungsweise geschrieben, wo zuletzt aufgehört worden ist – also, wo der Zeiger gerade steht.

**Versetzen**

Diese Funktion erlaubt es, Teilbereiche (auch ganze relative Dateien) innerhalb einer Datei zu versetzen beziehungsweise auf eine andere Datei zu übertragen. Zunächst müssen Anfang und Ende des Bereiches angegeben werden, der versetzt werden soll. Danach muß die Zieldatei angegeben werden. Soll innerhalb der relativen Datei versetzt werden, so muß als Zieldatei einfach der Name der Quelldatei (= Datei, mit der gerade gearbeitet wird) angegeben werden.

Letztendlich muß nur noch der Anfang des neuen Bereiches angegeben werden. Beachten Sie bitte, daß die Satzlänge der Zieldatei entweder größer oder mindestens genauso groß wie die der Quelldatei sein muß!

**Ordnen**

Heißt nichts anderes als Sortieren einer Datei. Zunächst wird der Bereich angegeben, der sortiert werden soll (darf nicht größer als 300 Sätze sein!). Dann muß markiert werden, welcher Teilbereich beziehungsweise welcher Schlüssel der Datei als Sortierkriterium gelten soll. Mittels den Cursortasten LEFT-RIGHT wird der Bereich markiert. Als Kriterium gilt der Bereich, unter dem Einser stehen. Mit den Tasten 1 und 0 können solche Einser gesetzt beziehungsweise gelöscht werden. Beispiel:

```
001BETRACHTEN  56
00111111111111100
```

Nun gilt zum Beispiel bei einer Verben-Tabelle, wie wir sie schon bald kennenlernen werden, das Verb als Kriterium.

Zur Sortierroutine, die im Editor integriert ist, ist zu bemerken, daß sie nur für kleine Bereiche geeignet ist, deren Satzlänge kleiner als 40 Zeichen ist. In einem anderen Kapitel dieses Kurses finden Sie noch ein Sortierprogramm für riesige Dateien mit Satzlängen bis zu 255 Zeichen!

**Drucken**
Durch Aufruf dieser Routine können relative Dateien beziehungsweise Teilbereiche einer Datei mit einem Drucker ausgegeben werden. Das Druckermenü bietet dann folgende Auswahlmöglichkeiten:
Normale Ausgabe: Der Text wird so ausgegeben, wie er mit dem normalen Zeichensatz des C64 aussieht.
D.z.s.-Ausgabe: Ein Text, der mit Hilfe des deutschen Zeichensatzes erstellt wurde, wird umgerechnet ausgegeben. Das heißt, anstelle von ß wird ss, anstelle von ä ein ae ausgedruckt etc. Dies ist nötig, da die Druckerroutine für einen MPS-801 »Billigdrucker« geschrieben wurde.
Eigene Routine: Wenn Sie einen besseren Drucker haben (Neid!!!), können Sie sich eine eigene Druckroutine schreiben (siehe Listingsdokumentation). Wer will, kann mittels Bit-Gefummel den deutschen Zeichensatz auch auf einem MPS-801 darstellen.
**Inhaltsverzeichnis**
Hier wird das Directory ohne Programmverlust ausgegeben.
**Zentrieren**
Durch diese Routine können Dateibereiche zentriert werden. Es ist so möglich, den Text formatiert auszugeben. Es gibt zwei Formatierungsmöglichkeiten: Links-rechtsbündig zentriert und in die Mitte zentrieren. Bei ersterem werden einfach soviele SPACEs eingefügt, bis der Text am linken und rechten Rand anstößt. Beim in die Mitte zentrieren wird jeder Satz so dargestellt, daß links und rechts gleich viele SPACEs sind.
**Fehlerkanalabfrage:**
Hiermit kann der Fehlerkanal des Floppy-Laufwerks abgefragt werden.
**Ende**
Muß unbedingt betätigt werden, wenn man die Arbeit an einem Text beenden will. Nur dann wird gewährleistet, daß der Puffer des Diskettenlaufwerks auf Disk gespeichert wird und so keine Daten verloren gehen können.
Sie können den Editor auch jederzeit mittels RUN/STOP-Taste verlassen. Mittels GOTO 50 im Direktmodus kommen Sie dann wieder ins Hauptmenü.
Dies ist auch schon alles, was Sie zum Umgang mit dem Editor wissen müssen. Wir werden im Rahmen des Kurses den Editor noch oft benötigen. Ich rate Ihnen deshalb, sich schon jetzt mit seinen einzelnen Funktionen gründlich vertraut zu machen. Im nächsten Kapitel finden Sie ein weiteres Utility, das ebenfalls (wie der Zeichensatz-Editor) direkt auf den Text-Editor für relative Dateien anwendbar ist: Ein Sortierprogramm für relative Dateien.

## Das Sortieren von riesigen Datenfeldern (relativen Dateien)

Bisher haben wir immer nur kleine Tabellen sortiert. Also Dateien, die immer komplett im Computer standen. Unser Texteditor für relative Dateien ist zwar in der Lage, riesige Dateien in den Griff zu bekommen, was jedoch das Sortieren von Datenfeldern anbelangt, so bedient er sich eines kleinen Kunstgriffes. Der Dateiabschnitt, der sortiert werden soll, wird komplett in ein Puffer-Stringfeld innerhalb des Basic-RAMs geladen. Dieser Puffer wird sodann mit Hilfe eines Quicksort-Unterprogramms sortiert. Anschließend wird der Pufferinhalt wieder auf die relative Datei auf Disk übertragen. Die »Sortierkapazität« unseres Texteditors ist also auf die Größe des Puffers und somit auf den Speicher des C64 begrenzt. Dieser Puffer kann nicht als besonders groß bezeichnet werden. Sie werden mit Sicherheit sehr bald auf seine Grenzen stoßen. Spätestens dann wird es kritisch, wenn man den Wortschatz zu einem Adventure sortieren will. Wir haben schließlich vor, Wortschätze mit über 1000 Worten zu verwalten – und die wollen erst einmal sortiert sein.
Wir brauchen also ein Sortierprogramm, das in der Lage ist, Dateien von Disk zu sortieren, die nicht auf einmal in den Speicher unseres C64 geladen werden können.
Das Programmieren von solchen Datei-Sortier-Programmen ist eine äußerst haarige Sache. Auch die besten Programmierer müssen sich vor dem Programmieren lange und genaue Überlegungen und Skizzen auf Papier zeichnen, um genauen Durchblick zu behalten. Die Schwierigkeit liegt weniger bei der komplizierten Programmiertechnik, die erforderlich ist, sondern vielmehr bei den Grundüberlegungen, die gemacht werden müssen. Ähnlich wie bei den einzelnen Sortieralgorithmen (Vgl. Kurs über effektives Programmieren in der Zeitschrift 64'er) gibt es auch hier zahlreiche Möglichkeiten zum Lösen des Problems. Die Grundroutinen sind zwar immer die gleichen, es gibt jedoch unendlich viele Möglichkeiten, sie zu kombinieren. Auch die besten Sortiermethoden für Dateien, die komplett auf einmal im Speicher sind, setzen sich aus mehreren Sortieralgorithmen zusammen (zum Beispiel Quicksort und Bubblesort kombiniert), um eine möglichst schnelle Ausführungszeit zu erreichen. Bei diskorientierten Sortierprogrammen spielt neben dem Zeitaufwand, der zum Sortieren gebraucht wird, auch noch ein anderer Faktor eine dominierende Rolle: Die Zahl der Zugriffe auf Disk. Selbstverständlich ist es möglich, unser Quicksort-Unterprogramm (Listing 8) mit wenig Aufwand so zu modifizieren, daß es in der Lage ist, eine riesige Datei auf Disk zu sortieren. Dabei wären bei großen Dateien zehntausende, ja bis zu über 100 000 Schreib-Lese-Operationen mit dem Diskettenlaufwerk notwendig. Sie werden mir sicherlich zustimmen, wenn ich nun behaupte, daß eine solche Sortiermethode eine unglaublich große Strapaze für unser Floppy-Laufwerk (ganz zu schweigen von der Diskette, die dieser Tortur ausgesetzt wird) wäre.
Eine andere Methode wäre es zum Beispiel, mehrere Disketten zu verwenden. Zum Beispiel die Quelldisk, auf der sich die Tabelle befindet und eine weitere Disk, auf der das Sortierprogramm zwischenspeichern kann. Dies wäre zumindest schonend für die Diskette.
Beide Methoden sind jedoch äußerst zeitaufwendig. Eine Methode, möglichst schnell zu sortieren, ist es, im Speicher des C64 nur die Satznummern zu puffern und diese zu sortieren, um dann gemäß dieser sortierten Satznummern die ganze Datei umzuordnen. Während man mit der allereinfachsten Methode (gesamte Datei auf einmal im Speicher) etwa 1000 Sätze (mittlerer Länge) sortieren kann, erlaubt die zuletzt vorgestellte Methode schon ein Vielfaches dieser Kapazität.
Die Methode hat jedoch wieder eine Einschränkung: Man kann eine bestimmte Satzanzahl (Größe des Satznummern-Puffers im Speicher) nicht überschreiten.
Wir wollen keine halben Sachen machen und ein absolut endgültiges Sortierprogramm schreiben, das folgende Eigenschaften haben soll:
- Es kann jede Datei, die sich auf einer Diskettenseite in Form einer relativen Datei unterbringen läßt, sortiert werden.
- Die Sortierung muß möglichst Disketten- und Floppy-Laufwerk-schonend ablaufen. Auf schnelle Ausführungszeit wird deshalb weniger Wert gelegt.

Auf den folgenden Seiten werde ich Ihnen anhand von Ablaufplänen und Schemen ausführlich erläutern, wie man eine solche Sortierung programmiert. Diese Grundlagen sind übrigens in keinster Weise auf den C64 beschränkt – sie gelten auch für jedes andere Computersystem, mit dem relative Dateien verwaltet werden können.
Lehnen Sie sich also gemütlich in Ihren Sessel, besorgen Sie sich was zu knabbern und/oder trinken und studieren Sie ausführlich die folgenden Seiten!

| SATZ | INHALT |
|---|---|
| 01 | NIMM |
| 02 | VERLIER |
| 03 | SUCHE |
| 04 | GEBE |
| 05 | LIES |
| 06 | AUF |
| 07 | ZU |
| 08 | BETE |
| 09 | SAGE |
| 10 | ESSE |
| 11 | HOLE |
| 12 | INVENTUR |
| 13 | BERATE |
| 14 | TOETE |
| 15 | BITTE |

**Bild 6a.**
**Ausgangstabelle unsortiert**

Im folgenden finden Sie eine Reihe von Diagrammen, die die Sortierung relativer Dateien erläutern.

| INHALT | SATZ |
|---|---|
| | |
| | |
| | |
| | |
| | |
| | |

**Bild 6b.**
**Datenpuffer der Kapazität M%**

ANFANGZEIGER

| | |
|---|---|
| 01 | NIMM |
| 02 | VERLIER |
| 03 | SUCHE |
| 04 | GEBE |
| 05 | LIES |
| 06 | AUF |
| 07 | ZU |
| 08 | BETE |
| 09 | SAGE |
| 10 | ESSE |
| 11 | HOLE |
| 12 | INVENTUR |
| 13 | BERATE |
| 14 | TOETE |
| 15 | BITTE |

| | |
|---|---|
| AUF | 06 |
| BERATE | 13 |
| BETE | 08 |
| BITTE | 15 |
| ESSE | 10 |
| GEBE | 04 |

**Bild 6c.**
**Die M%-niedrigsten Elemente der unsortierten Ausgangsdatei (Zeiger auf 1. Satz) werden in den Puffer gelesen und dabei gleichzeitig alphabetisch geordnet (im Puffer!).**
**Die Daten, die in den Puffer gelesen werden (bzw. die Sätze), werden gleichzeitig in der Ausgangsdatei gelöscht.**

Anhand der folgenden Skizzen wollen wir uns zunächst einmal grundlegende Gedanken über das Sortieren von riesigen Dateien machen. Das Verstehen der Programmiertechnik wird Ihnen dann viel leichter fallen. Zunächst benötigen wir wieder eine Ausgangsdatei. Um es möglichst einfach zu halten, beschränke ich mich hier auf eine Datei mit nur einem Schlüssel: Eine Verben-Datei, wie sie uns im Laufe des Kurses noch oft in ähnlicher Form begegnen wird. Diese Beispielsdatei sehen Sie in Bild 6a. Die »Ausgangsdatei« besteht aus 15 Sätzen – Verben in total ungeordneter Reihenfolge. Wir wollen einmal annehmen, daß der Speicher unseres C 64 viel zu klein ist, um diese Datei auf einmal fassen zu können – beziehungsweise die Datei zu groß ist, um komplett in den Computer geladen werden zu können. Wir benötigen also einen Puffer – ein Stringfeld, in das wir Teile der riesigen Datei laden können, um sie dort zu verändern. Das folgende Beispiel macht besonders leicht deutlich, wie so ein »Puffer« arbeitet.

Herr Helm hat zwei große Schwimmbecken – eines leer und eines voll mit Wasser. Er möchte das leere Schwimmbecken nun mit dem Wasser des vollen Schwimmbeckens füllen. Da er das volle Becken jedoch nicht einfach hochheben, zum anderen Becken hintragen und hineinleeren kann, braucht er ein Hilfsmittel. Zum Beispiel einen Eimer, der maximal so groß sein darf, daß Herr Helm ihn im gefüllten Zustand noch gut tragen kann. Nun kann er den Eimer im vollen Becken mit Wasser vollschöpfen, zum leeren Becken gehen und den Eimer dort hineinleeren. Herr Helm muß also mehrmals zwischen den Becken hin und her laufen, bis das volle Becken leer und das vorher leere Becken voll ist. Je kleiner der Eimer ist, umso öfter muß Herr Helm zwischen den Becken hin und her gehen...

Der Eimer ist als »Puffer« beziehungsweise »Datenpuffer« zu sehen. Den »Eimer«, den wir als Puffer zum Sortieren benutzen werden, sehen Sie in Bild 6b. Er faßt bis zu 6 Sätze. In Wirklichkeit ist der Speicher des C64 natürlich groß genug, um mehrere hundert Sätze puffern zu können. Der Einfachheithalber wollen wir uns jedoch auf einen kleinen Puffer beschränken. Unser Puffer besteht aus zwei Teilen. Im einen (M$(1)–M$(%)) werden die Sätze gespeichert. M% ist die Größe des Puffers – also 6. In der anderen Hälfte des Puffers (M$(1)–M$(M%)) wird gespeichert, an welcher Stelle die Sätze in der relativen Datei stehen – also die Nummern der Sätze.

Als Ziel des gesamten Sortiervorgangs soll die Verben-Datei alphabetisch geordnet vorliegen. Wie beginnen wir nun den Sortiervorgang? Das einzige Feld, das wir sortieren können, ist der Puffer. Wir müssen also zuerst Sätze von der Verben-Datei in den Puffer lesen, um sie dort verarbeiten zu können. Unsinnig wäre es, einfach die ersten 6 Sätze der Datei in den Puffer zu lesen und zu sortieren – das bringt nichts.

Wir durchlaufen zunächst die ganze Verben-Datei einmal von Anfang bis Ende und suchen die niedrigsten 6 Sätze heraus. Diese Sätze schreiben wir in den Puffer (Schema Bild 6c).

Das Suchschema läuft so ab, daß die 6 Sätze am Ende bereits alphabetisch im Puffer abgelegt sind. Gleichzeitig werden alle Sätze, die in den Puffer geschrieben werden, auf der Ausgangsdatei gelöscht. Nachdem der Puffer voll ist, sieht die relative Datei auf Disk wie in Bild 6d aus:

Betrachten wir nun einmal die ersten M%-Sätze der jetzigen Datei. An genau dieser Stelle müssen am Ende des Sortiervorgangs die 6 niedrigsten Sätze stehen. Diese Sätze befinden sich in unserem Puffer. Allerdings können wir den Puffer jetzt nicht gleich dorthin übertragen, da sich in ihm noch volle Sätze befinden, die beim Zurückschreiben des Puffers überschrieben werden würden – also verlorengingen.

Zählen wir doch einmal, wie viele volle Sätze noch im oberen Bereich stehen: genau 4 (Satz 1, 2, 3 und 5). Wir müssen also 4 Sätze »in Sicherheit« bringen, bevor der Pufferinhalt in den oberen Bereich geschrieben werden darf.

Aber wohin denn nun mit den 4 Sätzen? Sicherlich werden Sie schon festgestellt haben, daß sich im unbetroffenen Teil der Datei, der beim Übertragen des Puffers nicht überschrieben wird, genau 4 leere Stellen befinden. 4 Stellen, an denen vorher Sätze standen, die sich jetzt gerade im Puffer befinden und die später in den oberen Bereich geschrieben werden. Es bietet sich also an, die 4 Sätze im oberen Bereich in die 4 Leersätze des unteren Bereiches zu schreiben (s. Bild 6e).

Wenn Sie nicht ganz mitgekommen sind, lesen Sie bitte den Abschnitt noch mal durch. Es gibt nun verschiedene Methoden, den oberen Bereich zu räumen. Die einfachste ist, die gesamte Datei mit Schleifen zu durchlaufen. Dies erfordert jedoch wieder zahlreiche Lesezugriffe per Disk und wir haben uns vorgenommen, ein möglichst Disketten- und

| | |
|---|---|
| 01 | NIMM |
| 02 | VERLIER |
| 03 | SUCHE |
| 04 | |
| 05 | LIES |
| 06 | |
| 07 | ZU |
| 08 | |
| 09 | SAGE |
| 10 | |
| 11 | HOLE |
| 12 | INVENTUR |
| 13 | |
| 14 | TOETE |
| 15 | |

**Bild 6d. Ausgangsdatei nach dem 1. »Füllen« des Puffers.**

| | |
|---|---|
| 01 | |
| 02 | |
| 03 | |
| 04 | |
| 05 | |
| 06 | |
| 07 | NIMM |
| 08 | VERLIER |
| 09 | SUCHE |
| 10 | LIES |
| 11 | ZU |
| 12 | SAGE |
| 13 | HOLE |
| 14 | INVENTUR |
| 15 | TOETE |

**Bild 6e. Unser nächstes Ziel: Platz schaffen, um den Puffer wieder in die Ausgangsdatei zurückschreiben zu können.**

| | | | |
|---|---|---|---|
| AUF | 06 | | 04 |
| BERATE | 13 | | 06 |
| BETE | 08 | ORDNEN | 08 ← PUFFER-ZEIGER |
| BITTE | 15 | | 10 |
| ESSE | 10 | | 13 |
| GEBE | 04 | | 15 |

Jetzt »freie« Sätze in der Ausgangsdatei.

Jetzt »leere« Sätze in geordneter Reihenfolge.

**Bild 6f. Hier sehen Sie, wie nach dem Aussortieren der ersten Sätze Platz zum Zurückschreiben geschaffen wird.**

Laufwerk-schonendes zu entwickeln. Wir entscheiden uns deshalb für einen komplizierteren Weg, der jedoch nur sehr wenige Zugriffe mit dem Floppy-Laufwerk erfordert.

Wie ich bereits sagte, ist der Puffer in zwei Hälften unterteilt - eine für die Sätze selbst und eine für die Satznummern. Werfen Sie bitte einmal einen Blick auf Bild 6f. Hier wird gezeigt, wie die Pufferhälfte, in der die Satznummern stehen, mit der kleinsten Satzzahl beginnend sortiert wird. Hierfür verwenden wir unser Quicksort-Unterprogramm, das aus Geschwindigkeitsgründen jedoch auf Zahlen-Sortieren modifiziert wird.

Nachdem die Nummern-Pufferhälfte sortiert ist, wird ein Pufferzeiger gesetzt - der genau auf den ersten leeren Satz zeigt, der sich nicht mehr innerhalb des oberen Bereiches befindet (in unserem Beispiel SATZ 8). Dies ist nötig, damit versehentlich nicht Daten, die aus dem oberen Bereich »gerettet« werden sollen, in einen leeren Satz innerhalb des oberen Bereiches geschrieben werden.

Bild 6g zeigt schließlich, wie die 4 Sätze aus dem oberen Bereich auf die 4 leeren Stellen des unteren Bereiches verteilt werden. Nun kann der Pufferinhalt wieder in die Verben-Datei zurückgeschrieben werden (Bild 6h). Die ersten M%-Sätze der Datei sind damit alphabetisch geordnet.

Nun wird der Zeiger der Verben-Datei, der bisher an erster Stelle stand (weil wir die ganze Datei und nicht nur einen bestimmten Bereich sortieren wollen) um den Wert M% erhöht. Er zeigt sodann auf die Stelle, an der der noch unsortierte Bereich der Datei beginnt (Bild 6i). Der nächste Sortierabschnitt kann nun beginnen.

Sind alle Sortierabschnitte beendet - also jeweils Puffer mit den M% niedrigsten Elementen vom unsortierten Bereich füllen, Sätze aus dem Pufferbereich in Sicherheit bringen, Puffer zurückschreiben etc. - liegt die komplette Verben-Datei sortiert vor (Bild 6j).

## Eine Frage ist noch offen ...

Woher das BIER in SATZ 16 kommt? Nun, das haben Sie sich nach diesem Brocken Theorie mit Sicherheit verdient (und ich auch!)...

Anschließend finden Sie das Listing 24 zum Sortierprogramm. Bitte studieren Sie auch die Dokumentation ausführlich, damit Sie später in der Lage sind, das Programm eigenen Vorstellungen entsprechend zu ergänzen beziehungsweise abzuändern.

**Dokumentation zum Super-Sort-Listing**

**100 - 130** Hier werden die Felder für die beiden Pufferteile und den Quicksort definiert. Durch Ändern der Variable M% kann die Größe des Puffers geändert werden (momentaner Wert 300).

**200 - 295** Anfangsmenü und Eingabe aller Parameter. AN - EN, Bereich innerhalb der Datei NS, der sortiert werden soll. ZL, ZR begrenzen den Bereich innerhalb der Sätze, nach dem sortiert werden soll. Beispiel: MID$(A$,ZL,ZR)

**300 - 330** Vorbereitung für den ersten Sortiervorgang (Sortieren der M% niedrigsten Sätze, beginnend ab dem Zeiger auf die Ausgangsdatei). ZD, Zeiger in Datei. Um die M% niedrigsten Sätze zu finden, wird die gesamte Datei von Satz ZD bis zum letzten Satz durchlaufen. VG$ wird als Vergleichsargument bestimmt.

**400 - 470** Hier wird mittels einer Schleife die Datei ab ZD bis Ende nach den M% niedrigsten Sätzen durchsucht. Dabei wird jeweils der Vergleichsstring VG$ mit einem Satz des Suchbereiches verglichen. Wird ein Satz gefunden, der kleiner als VG$ ist, so nimmt VG$ dessen »Wert« an. In der Variablen MM wird gespeichert, an welcher Stelle der Datei der momentane »Wert2«, der in VG$ gespeichert ist, steht. Ist die Datei einmal durchlaufen, so steht in VG$ der kleinste Satz des durchsuchten Bereiches.

**500 - 570** Der Satz MM, also die zugehörige Satznummer zum im Durchlauf gefundenen, niedrigsten Wert VG$, wird in der Datei gelöscht. Löschen ist eigentlich eine falsche Bezeichnung. In den Satz MM wird vielmehr ein »%« geschrieben. Dies ist notwendig, da ein leerer Satz, also A$="" schließlich auch ein Satz ist. Leere Sätze sind kleiner als alles andere. Hat eine Datei, die sortiert wird, leere Sätze, so stehen diese nach dem Sortiervorgang immer an den ersten Stellen in der geordneten Datei. In der Datei, die sortiert werden soll, darf somit nie in einem Satz ein einzelnes »%« stehen, da es dann zu

| | |
|---|---|
| 01 | NIMM |
| 02 | VERLIER |
| 03 | SUCHE |
| 04 | |
| 05 | LIES |
| 06 | |
| 07 | ZU |
| 08 | |
| 09 | SAGE |
| 10 | |
| 11 | HOLE |
| 12 | INVENTUR |
| 13 | |
| 14 | TOETE |
| 15 | |

**Bild 6g. Die Sätze, die noch in dem Bereich stehen, in den der Puffer übertragen werden soll, werden in »Sicherheit« gebracht (in den unteren Teil der Ausgangsdatei).**

| | |
|---|---|
| 01 | AUF |
| 02 | BERATE |
| 03 | BETE |
| 04 | BITTE |
| 05 | ESSE |
| 06 | GEBE |
| 07 | ZU |
| 08 | NIMM |
| 09 | SAGE |
| 10 | VERLIER |
| 11 | HOLE |
| 12 | INVENTUR |
| 13 | SUCHE |
| 14 | TOETE |
| 15 | LIES |

**Bild 6h. Der Puffer ist in die Ausgangsdatei zurückgeschrieben worden. Die ersten M%-Elemente (Sätze) sind jetzt alphabetisch geordnet. Die restliche Datei ist noch unsortiert.**

| | | |
|---|---|---|
| | 01 | AUF |
| | 02 | BERATE |
| | 03 | BETE |
| | 04 | BITTE |
| | 05 | ESSE |
| | 06 | GEBE |
| Zeiger → | 07 | ZU |
| | 08 | NIMM |
| | 09 | SAGE |
| | 10 | VERLIER |
| | 11 | HOLE |
| | 12 | INVENTUR |
| | 13 | SUCHE |
| | 14 | TOETE |
| | 15 | LIES |
| | 16 | BIER |

**Bild 6i. Natürlich müssen jetzt auch noch die restlichen Elemente sortiert werden. Dazu wird einfach der Zeiger um den Wert M% erhöht. Die nächste Sortierrunde kann losgehen.**

Schwierigkeiten beim Lückenfüllen kommen kann. Wer auf »%« in seiner Datei nicht verzichten kann, kann im Listing das »%« auch durch ein anderes Zeichen ersetzen, das dann nicht verwendet werden kann (in der Datei). Nun wird geprüft, ob der Puffer schon voll ist (Zeile 540). ZP ist der Pufferzeiger. Sein Wert wird immer dann um 1 erhöht, wenn ein Satz in den Puffer geschrieben worden ist. Ist noch Platz im Puffer, geht es so weiter: Ein neues Vergleichselement VG$ muß bestimmt werden. Dabei muß aufgepaßt werden, daß das neue VG$ niemals " " sein darf. Hierfür sorgt Zeile 560. Nun folgt ein Rücksprung zu 400 – es wird wieder das kleinste Element gesucht.

**600 – 670** Der Puffer ist jetzt voll. Die Pufferhälfte MS(1)–MS(M%), in der die Satznummern gespeichert sind, wird nun mittels dem Quicksort-Unterprogramm sortiert (vergleiche Bild 6). Anschließend wird der Pufferzeiger ZP auf den ersten Satz gesetzt (Leersatz beziehungsweise Satz mit »%«), der im unteren Bereich liegt (Zeile 650 – 670).

**700 – 760** Hier werden die vollen Sätze, die noch immer im oberen Bereich stehen, evakuiert – also in die leeren Sätze des unteren Bereiches geschrieben. Hierzu wird der obere Bereich mit einer Schleife durchlaufen. Wird ein voller Satz gefunden, so wird er in den Leersatz geschrieben, auf den der Pufferzeiger gerade deutet. Der Pufferzeiger wird sodann um den Wert 1 erhöht.

**800 – 860** Der Puffer wird mittels Schleife in den oberen Bereich ab ZD (Zeiger auf Datei) geschrieben.

**900 – 950** Nun wird der nächste Sortierabschnitt vorbereitet. Der Dateizeiger ZD wird um den Wert M% erhöht – also auf den Satz gestellt, bei dem der noch unsortierte Bereich der Datei beginnt. Ein neues Vergleichs-Element wird bestimmt. Der Pufferzeiger wird wieder auf den Wert 1 zurückgesetzt.

**Listing 24. Das Sortierprogramm für relative Dateien ▶**

```
0 REM ********************************       <050>
1 REM *    SORTIERPROGRAMM FUER       *       <250>
2 REM *      RELATIVE DATEIEN         *       <250>
3 REM *       <C> 1985 BEI            *       <122>
4 REM *      MICHAEL NICKLES          *       <121>
5 REM ********************************       <055>
100 M%=300     :REM GROESSE DES PUFFERS      <173>
105 N%=M%      :REM WERT VON M% SPEICHERN    <235>
110 DIM M$(M%) :REM PUFFER FUER SAETZE       <198>
120 DIM MS(M%) :REM PUFFER FUER ZAHLEN       <024>
130 DIM LG(100),RG(100):Z=0:LG(1)=1:RG(1)=
    M%           :REM FUER QUICKSORT         <088>
200 REM ANFANGSMENUE -------------------
    ------------------------------------     <095>
201 :                                        <177>
205 POKE 53280,0:POKE 53281,0:PRINT"{CLR,W
    HITE}";                                  <249>
210 PRINT"{RVSON,6SPACE}REL.DATEI SORTIERP
    ROGRAMM{9SPACE,RVOFF}";                  <216>
215 PRINT"{RVSON,9SPACE}VON MICHAEL NICKLE
    S{12SPACE,RVOFF}"                        <040>
220 INPUT"{2DOWN}DATEINAME ";N$              <226>
230 OPEN 15,8,15:REM FLOPPYKANAL AUF         <169>
235 PRINT#15,"I0":REM INITIALISIEREN         <138>
240 OPEN 1,8,3,N$ :REM REL.DATEI OEFFNEN     <174>
250 INPUT#15,A,B$:REM FEHLERABFRAGE          <235>
254 IF A<>0 THEN PRINT"{RVSON}ERROR{RVOFF,
    2SPACE}";B$:END                          <189>
270 PRINT"{2DOWN,8SPACE,RVSON}G{RVOFF}ANZE
     DATEI SORTIEREN                         <172>
275 PRINT"{DOWN,8SPACE,RVSON}T{RVOFF}EILBE
    REICH SORTIEREN                          <060>
277 PRINT"{DOWN,8SPACE,RVSON}L{RVOFF}ESEN    <175>
280 GET I$:IF I$<>"G"AND I$<>"T"AND I$<>"L
    "THEN 280                                <032>
281 IF I$="L"THEN 10000                      <164>
285 IF I$="G"THEN AN=1:GOTO 290              <073>
286 PRINT"{DOWN}BITTE GEBEN SIE ANFANG UND
     ENDE DES{5SPACE}BEREICHES AN, DER SOR
    TIERT";                                  <006>
287 PRINT" WERDEN SOLL!"                     <185>
288 INPUT"ANFANG ";AN                        <038>
289 INPUT"ENDE{3SPACE}";EN:IF AN>EN THEN P
    RINT"UNSINN!":GOTO 295                   <126>
290 PRINT"{DOWN}BITTE GEBEN SIE AN, NACH W
    ELCHEM STRING-BEREICH DIE SORTIERUNG     <180>
291 PRINT"ERFOLGEM SOLL!"                    <135>
292 INPUT"{DOWN}VON ZEICHEN ";ZL             <053>
293 INPUT"BIS ZEICHEN ";ZR                   <002>
295 ZR=ZR-ZL+1                               <040>
300 REM SORTIERVORGANG ANFANG -----------
    ------------------------------------     <202>
301 :                                        <023>
305 TI$="000000"       :REM ZEIT AUF 0       <133>
310 ZD=AN              :REM ZEIGER SETZEN    <037>
```

| | |
|---|---|
| 01 | AUF |
| 02 | BERATE |
| 03 | BETE |
| 04 | BITTE |
| 05 | ESSE |
| 06 | GEBE |
| 07 | HOLE |
| 08 | INVENTUR |
| 09 | LIES |
| 10 | NIMM |
| 11 | SAGE |
| 12 | SUCHE |
| 13 | TOETE |
| 14 | VERLIER |
| 15 | ZU |
| 16 | BIER |

**Bild 6j. Nach einer langen Tour-de-Force ist unsere Datei sortiert. Bitte beachten Sie Datensatz 16. Die Erklärung finden Sie auf Seite 34.**

Die komplett sortierte Datei.

```
320 ZP=1                :REM PUFFERZEIGER      <108>
325 SA=ZD:GOSUB 9000:GOSUB 9100
                        :REM 1.SATZ LESEN <041>
330 VG$=A$              :REM VERGL.SATZ        <228>
400 REM DIE M% KLEINESTEN ELEMENTE VON SAT
    Z ZD BIS LETZTEN SATZ SUCHEN----------  <065>
401 :                                          <123>
402 PRINT"{DOWN,RVSON}";ZP;"{LEFT}. DURCHL
    AUF{4SPACE,RVOFF,DOWN}"                    <027>
410 SA=ZD+1:REM ----- BEGIN LOOP -----         <203>
411 PRINT SA,A$         :REM SUPERMIKE         <195>
430 :GOSUB 9000         :REM POSITIONIEREN     <134>
431 IF A=50 THEN M%=SA-ZD:EN=SA:GOTO 500
                                  :REM END
    LOOP                                       <058>
432 IF SA=EN+1 THEN 500
                        :REM END LOOP          <082>
435 :GOSUB 9100         :REM A$ LESEN          <149>
440 :IF MID$(A$,ZL,ZR)>MID$(VG$,ZL,ZR)OR A
    $="@"THEN 460                              <071>
445 : VG$=A$            :REM NEUER VG$         <209>
450 : MM=SA             :REM SATZNR. MERKEN    <014>
460 SA=SA+1             :REM NEXT SATZ         <035>
465 GOTO 411:REM ----- CONT LOOP  -----        <181>
470 :                                          <192>
500 REM KLEINSTES ELEMENT IM BEREICH ZD BI
    S ZD+M$ = VG$ WURDE GEFUNDEN ---------  <173>
501 :                                          <223>
510 SA=MM:GOSUB 9000:A$="@":GOSUB 9200
                        :REM SATZ MM LOESCH
    EN                                         <002>
520 M$(ZP)=VG$:MS(ZP)=MM
                        :REM PUFFER FUELLEN    <230>
530 ZP=ZP+1          :REM PUFFER-ZEIGER+1      <087>
540 IF ZP>M%THEN 600 :REM PUFFER VOLL          <230>
550 REM NEUEN VG$ SUCHEN                       <029>
555 SA=ZD                                      <013>
560 GOSUB 9000:GOSUB 9100:IF A$="@"THEN SA
    =SA+1:GOTO 560                             <176>
565 VG$=A$:MM=SA                               <193>
570 GOTO 400                                   <014>
600 REM PUFFER IST JETZT VOLL ------------
    --------------------------------------  <008>
601 :                                          <069>
610 PRINT"{DOWN,RVSON}PUFFERINHALT{RVOFF,D
    OWN}":FOR I=1 TO M%:PRINT M$(I),MS(I):
    NEXT I:REM SUPERMIKE                       <158>
620 REM MS(1)-MS(M%) WIRD JETZT MIT QUICKS
    ORT SORTIERT ---------------------------  <203>
621 :                                          <089>
625 GOSUB 30000:REM AUFRUF QUICKS.UP           <124>
630 PRINT"{DOWN,RVSON}SORTIERTER NUMMERPUF
    FER{RVOFF,DOWN}":FOR I=1 TO M%:PRINT M
    S(I):NEXT:REM SUPERMIKE                    <192>
650 REM NEUEN PUFFERZEIGER SETZEN ---------
    --------------------------------------  <174>
651 :                                          <119>
655 ZP=1                                       <207>
660 IF MS(ZP)<ZD+M%THEN ZP=ZP+1:GOTO 660       <089>
670 PRINT"ZP=";ZP        :REM SUPERMIKE        <204>
700 REM PLATZ FUER PUFFER SCHAFFEN -------
    --------------------------------------  <004>
701 :                                          <169>
710 FOR I=ZD TO ZD+M%-1                        <252>
720 :SA=I:GOSUB 9000:GOSUB 9100
                        :REM A$ LESEN          <155>
729 PRINT A$            :REM SUPERMIKE         <111>
730 :IF A$="@"THEN 760                         <187>
740 :   SA=MS(ZP):GOSUB 9000:GOSUB 9200
                        :REM LUECKE FUELLEN <155>
745 :   ZP=ZP+1         :REM P.ZEIGER +1       <136>
760 NEXT I                                     <080>
800 REM PUFFER ZURUECKSCHREIBEN ----------
    --------------------------------------  <099>
801 :                                          <015>
810 SA=ZD                                      <014>
820 : FOR I=1 TO M%                            <077>
830 :   A$=M$(I)                               <145>
831 :   PRINT A$            :REM SUPERMIKE     <088>
840 :   GOSUB 9000:GOSUB 9200                  <050>
850 :   SA=SA+1                                <196>
860 : NEXT I                                   <242>
900 REM NAECHSTEN SORTIERVORGANG VORBEREIT
    EN------------------------------------  <052>
901 :                                          <115>
910 IF M%<>N%OR O%THEN PRINT"SORTIERZEIT:
    ";TI$:PRINT"{2DOWN}*** FERTIG ***":CLO
    SE 15:END                                  <106>
920 ZD=ZD+M%:REM DATEIZEIGER ERHOEHEN          <217>
924 ZP=1      :REM PUFFERZEIGER AUF 1          <179>
925 SA=ZD                                      <129>
926 GOSUB 9000:GOSUB 9100:IF A$="@"THEN SA
    =SA+1:GOTO 560                             <032>
927 VG$=A$:MM=SA                               <045>
930 IF M%>EN-ZD THEN M%=EN-ZD                  <227>
935 IF M%=EN-ZD THEN O%=1                      <238>
950 GOTO 400 :REM NAECHSTE SORTIERRUNDE        <134>
9000 REM POSITIONIEREN ***************         <109>
9001 :                                         <087>
9010 : HB%=SA/256:LB%=SA-HB%*256               <005>
9020 : PRINT#15,"P";CHR$(3);CHR$(LB%);CHR$
     (HB%);CHR$(1)                             <125>
9030 : INPUT#15,A:REM FEHLERABFRAGE            <220>
9040 RETURN                                    <208>
9100 REM SATZ LESEN     ***************        <031>
9101 :                                         <187>
9110 : IF A=50 THEN RETURN: REM FEHLER         <068>
9120 : INPUT#1,A$                              <236>
9130 RETURN                                    <042>
9200 REM SATZ SCHREIBEN **************         <224>
9201 :                                         <031>
9210 : PRINT#1,A$                              <230>
9230 RETURN                                    <144>
10000 REM DATEI LESEN --------------------
      ------------------------------------
      --                                       <086>
10001 :                                        <071>
10010 PRINT"{RVSON}LESEN / ENDE{RVOFF}"        <114>
10015 GET I$:IF I$<>"L"AND I$<>"E"THEN 100
      15                                       <189>
10020 IF I$="E"THEN PRINT"{CLR}":GOTO 270      <195>
10030 INPUT"SATZNUMMER ";IA                    <103>
10034 PRINT"{RVSON}SATZ";IA;"{LEFT}-";IA+2
      0;"{RVOFF}"                              <047>
10035 FOR SA=IA TO IA+20                       <054>
10040 : GOSUB 9000:GOSUB 9100                  <098>
10050 : IF LEN(A$)<40 THEN PRINT A$            <103>
10060 : IF LEN(A$)>=40 THEN PRINT A$;          <015>
10070 NEXT SA                                  <092>
10080 GOTO 10010                               <202>
30000 REM  QUICKSORT ---------------------
      ------------------------------------
      --                                       <087>
30001 RG(1)=M%                                 <128>
30010 Z=Z+1:IF LG(Z)>=RG(Z)THEN 30120          <166>
30020 X=LG(Z):Y=RG(Z)                          <011>
30030 VG=MS(INT((X+Y)/2))                      <011>
30040 IF X>Y THEN 30100                        <205>
30050 IF MS(X)<VG THEN X=X+1:GOTO 30050        <153>
30060 IF MS(Y)>VG THEN Y=Y-1:GOTO 30060        <241>
30070 IF X>Y THEN 30100                        <235>
30080 S=MS(X):MS(X)=MS(Y):MS(Y)=S              <214>
30090 X=X+1:Y=Y-1:GOTO 30040                   <099>
30100 RG(Z+1)=Y:LG(Z+1)=LG(Z):GOSUB 30010      <184>
30110 LG(Z+1)=X:RG(Z+1)=RG(Z):GOSUB 30010      <104>
30120 Z=Z-1:RETURN                             <161>
```

**Listing 24. Das Sortierprogramm für relative Dateien**

Sodann wird überprüft, ob der Puffer größer ist als der Bereich, der noch sortiert werden muß. Ist dies der Fall, so wird der Puffer so verkleinert (M% in Zeile 930), daß der Bereich, der noch sortiert werden muß, genau in ihn hineinpaßt. In Zeile 910 wird solch eine Puffergrößenveränderung festgestellt – die Sortierung ist beendet. Die Variable erfüllt den gleichen Zweck dann, wenn alles glatt aufgeht, und M% nicht erniedrigt wird.

**9000 - 9230** Unterprogramme zum Positionieren, Lesen und Schreiben eines Satzes auf eine relative Datei.

**10000 - 10080** Kleine Lese-Routine zum Orientieren vor dem Sortieren.

**30000 - 30120** Quicksort-Unterprogramm

Ich möchte an dieser Stelle nochmals darauf hinweisen, daß das hier vorgestellte Sortierprogramm keineswegs das bestmögliche ist. Verbesserungen sind insbesondere in punkto Sätze aus dem oberen Bereich in die Leersätze übertragen, möglich. Bevor Sie anhand eigener Ideen Verbesserungen vornehmen, sollten Sie jedoch stets bedenken, ob diese Verbesserungen Disketten und Floppy-Laufwerk schonend genug behandeln – also möglichst wenige Schreib-Lese-Operationen ausführen.

**Arbeiten mit dem Super-Sort-Programm**

Nach dem Programmstart müssen Sie den Namen der Datei angeben, die Sie sortieren möchten (Disk einlegen!).

Danach haben Sie die Auswahlmöglichkeit unter den drei Menüpunkten »Ganze Datei sortieren«, »Teilbereich sortieren« und »Lesen«. »Lesen« funktioniert wie beim relativen Dateien-Editor, und dient hier lediglich zur Orientierung für Sie. Wählen Sie »Ganze Datei sortieren«, so wird die gesamte Datei sortiert. Das Satzende wird automatisch durch Abfragen des Floppykanals nach Fehler 50 ermittelt.

»Teilbereich ermitteln« erlaubt es, einen bestimmten Bereich der Datei zu ordnen. Vor dem Sortieren müssen Sie noch angeben, nach welchem Stringbereich die Sortierung erfolgen soll – also welcher Stringbereich verglichen wird. Diese Funktion ist besonders wichtig beim Sortieren von Dateien mit mehr als einem Schlüssel.

Alle Zeilen im Programm, die mit REM SUPERMIKE gekennzeichnet sind, können weggelassen werden (zur Geschwindigkeitserhöhung). Sie ermöglichen es, den ganzen Sortiervorgang auf dem Monitor mitzuverfolgen.

Bedenken Sie, daß das Sortieren von Dateien mit mehreren 1000 Sätzen sehr viel Zeit in Anspruch nimmt. Es lohnt sich dann kaum, beim Computer zu warten.

Geheimtip: Sortiervorgang vor dem Schlafengehen starten – am nächsten Morgen ist alles vorbei!

Wer Angst vor einer zu hohen Stromrechnung hat, kann ein Schalter-Interface an den User-Port anschließen und das Programm so modifizieren, daß sich der Computer samt Peripherie am Ende des Sortiervorganges selbst abschaltet!

**Eine bessere Lösung**

Unser Super-Port funktioniert einwandfrei. Wer nur ab und zu sortieren will, ist mit ihm gut bedient. Die folgende Sortiermöglichkeit ist eine wesentliche Verbesserung gegenüber unserer ersten. Wer die Handhabung von Puffern und Zeigern verstanden hat, wird auch mit der folgenden Theorie keine Probleme haben.

Unser Super-Sort 1 benötigt noch ziemlich viele Disk-Zugriffe, um den Puffer zu füllen. Es gibt jedoch eine wesentlich schnellere und schonendere Lösung:

**Theorie zum Super-Sort 2**

Auch hier wird wieder ein Puffer der Größe M% verwendet. Zu Beginn des Sortiervorgangs wird der Puffer einfach mit den ersten M%-Sätzen der Datei gefüllt. Wir suchen also **nicht** wie bei Super-Sort 1 die M% niedrigsten Sätze heraus.

Nun wird der gesamte Pufferinhalt mit dem Quicksort sortiert. Der nächste Schritt besteht darin, die gesamte Datei vom Zeiger beginnend bis zum Ende zu durchlaufen. Der Zeiger steht beim ersten Sortierabschnitt bei Satz M%, beim zweiten Durchlauf auf 2mal M% etc. Bei jedem eingelesenen Satz wird geprüft, ob dieser kleiner ist als der größte Satz im Puffer (steht beim sortierten Puffer an letzter Stelle). Ist dies nicht der Fall, wird der nächste Satz gelesen. Tritt aber nun der Fall ein, daß der Satz kleiner als der größte Satz des Puffers ist, passiert folgendes:

Wir nennen den eingelesenen Satz nun einmal A$. A$ muß nun irgendwie in den Puffer geschrieben werden. Anhand der binären Suchmethode (an anderer Stelle des Kurses beschrieben) ermitteln wir, wo im Puffer A$ eingefügt werden muß. Der letzte Satz im Puffer wird dadurch aus dem Puffer hinausgeworfen (wir müssen schließlich Platz für A$ schaffen). Damit er jedoch nicht verlorengeht, wird er wieder in die Datei zurückgeschrieben – und zwar an die Stelle, an der A$, das sich nun im Puffer befindet, gestanden hat.

Diese Methode hat den Vorteil, daß pro Pufferfüllung die Datei nur ein einziges Mal durchlaufen werden muß. Auch das Problem, Platz für den Puffer-Zurückschreibevorgang zu

```
10 REM *********************************     <060>
15 REM * PROGRAMM ZUM AENDERN VON      *     <000>
20 REM * DISKNAME UND ID               *     <035>
25 REM *********************************     <075>
30 PRINT"{CLR,RVSON}DISK-UTILITY 1{26SPACE
   ,RVOFF}";                                 <024>
35 PRINT"{2DOWN,5SPACE}DISK-NAME AENDERN{3
   SPACE}-{2SPACE}'1'                        <166>
40 PRINT"{5SPACE}DISK-ID{3SPACE}AENDERN{3S
   PACE}-{2SPACE}'2'                         <017>
42 PRINT"{5SPACE}ENDE{16SPACE}-{2SPACE}'3'   <241>
45 GET X$:IF X$<>"1"AND X$<>"2"AND X$<>"3"
   THEN 45                                   <205>
46 IF X$="3"THEN PRINT"{2DOWN}*** FERTIG *
   **":END                                   <033>
50 OPEN 15,8,15:REM FLOPPYKANAL OEFFNEN      <148>
55 PRINT#15,"I0":OPEN 1,8,3,"#":PRINT#15,"
   U1:3"0",18,0":IF X$="1"THEN 90            <205>
60 PRINT#15,"B-P:3,162":GET#1,A$,B$,C$,D$,
   E$:PRINT"{DOWN}ALTER DISK-ID IST{SPACE,
   RVSON,SPACE}";A$;                         <070>
65 PRINT B$;C$;D$;E$:INPUT"{DOWN}NEUER DIS
   K-ID ";F$                                 <070>
70 IF LEN(F$)<>4 THEN F$=LEFT$(F$+"{6SPACE
   }",5)                                     <097>
75 PRINT#15,"B-P:3,162":PRINT#1,F$;:PRINT#
   15,"U2:3"DV",18,0                         <118>
80 I$="I"+MID$(STR$(DV),2):PRINT#15,I$       <072>
85 RUN                                       <127>
90 PRINT#15,"B-P:3,144":R$="":FOR L=1 TO 1
   6:GET#1,A$:R$=R$+A$:NEXT L                <029>
95 PRINT"{DOWN}ALTER DISK-NAME IST ";"{RVS
   ON}"R$"{RVOFF}"                           <089>
100 INPUT"{DOWN}NEUER DISK-NAME ";NE$        <148>
105 NE$=LEFT$(NE$+"{14SPACE}",16)            <248>
110 PRINT#15,"B-P:3,144":PRINT#1,NE$;:PRIN
    T#15,"U2:3"DV",18,0":GOTO 80             <098>
```

**Listing 25. Disk-Utility**

```
51200 REM UP - DISK EINLEGEN -----------     <123>
51210 PRINT"{RVSON}BITTE DISK-SEITE";SIDE;
      "{LEFT,SPACE}EINLEGEN !{RVOFF}":POKE
       198,0:WAIT 198,1                      <174>
51240 OPEN 2,8,2,"#":PRINT#15,"U1 2 0 18 0
      ":PRINT#15,"B-P";2;162:GET#2,I$        <089>
51250 CLOSE 2:IF SIDE<>VAL(I$)THEN 51200     <090>
51260 RETURN                                 <007>
```

**Listing 26. Ein nützliches Unterprogramm**

schaffen, entfällt. Bei Methode 2 wird zwar wesentlich mehr innerhalb des Rechners sortiert, es wird jedoch soviel Zeit bei Diskoperationen gespart, daß Methode 2 wesentlich schneller und Disk-schonender ist.

Wie gesagt, für diesen Kurs reicht Super-Sort I vollkommen aus. Wer viel sortieren will, sollte sich jedoch ein Sortierprogramm der oben beschriebenen Methode schreiben.

## Trick am Rande: Diskettenseite abfragen

An dieser Stelle möchte ich einmal ein Problem abhandeln, das immer dann auftaucht, wenn man Programme schreiben will, die sich über zwei oder mehrere Disk-Seiten erstrecken. Bei Adventure-Spielen und KI-Programmen, die sich selbst verwalten, ist dies besonders häufig der Fall. Im Programm kommt dann früher oder später die Anweisung BITTE DISKSEITE X EINLEGEN. Wie kann man nun innerhalb eines Programmes feststellen, ob auch die richtige Disk-Seite eingelegt worden ist? Dem Anfänger bereitet dies oft Schwierigkeiten, da Probleme wie diese nicht im Bedienungshandbuch des Floppy-Laufwerks behandelt werden. Die Lösung liegt einfach darin, den ID der eingelegten Disk einzulesen. Dies setzt natürlich voraus, daß der ID der ersten Diskette 01, der der zweiten 02 etc. beträgt. Im folgenden finden Sie deshalb zunächst ein kleines Hilfsprogramm, mit dem es möglich ist, Diskname und ID einer Disk nachträglich zu verändern ohne, daß die Disk formatiert werden muß beziehungsweise ein sich auf ihr befindliches Programm verloren geht (Listing 25).

Beachten Sie bitte, daß der Diskname nicht mehr als 17 und der Disk ID nicht mehr als 5 Zeichen Länge haben darf. Nun brauchen wir nur noch ein kleines Basic-Unterprogramm, mit dem der Disk-ID gelesen werden kann (Listing 26).

Bevor das Programm mittels GOSUB 51200 aufgerufen werden kann, muß der Floppykanal 15 geöffnet sein. Anschließend muß noch durch die Variable SIDE (beziehungsweise SI) bestimmt werden, welche Seite eingelegt werden soll. Das Unterprogramm führt erst dann ein RETURN aus, wenn die gewünschte Seite eingelegt ist. Innerhalb eines Basic-Programmes sieht dies dann so aus:

```
10 OPEN15,8,15
20 SIDE=2
30 GOSUB51200
40 END
```

Es bietet sich an, den ID der ersten Seite als 1SIDE, den der zweiten Seite als 2SIDE etc. zu benennen, da eine einzige Ziffer beim ID nicht ausreicht!

Vergessen Sie auch nicht aus Sicherheitsgründen, nach einem Disk-Wechsel immer neu zu initialisieren.

Sie können dieses Unterprogramm leicht in eigenen Basic-Programme einbauen. Das Disk- und ID-Änder-Programm speichern Sie am besten wieder auf Ihrer Utility-Disk, auf der sich auch die Editoren befinden. (M. Nickles/bs)

Bitte ein Adventure-Spiel!

?

# Künstliche Intelligenz

**Das in letzter Zeit so oft benutzte Schlagwort KI gilt nicht nur für Großcomputer. Machen Sie Ihren C 64 zur Denkmaschine!**

Hier beginnt der zweite große Hauptteil des Kurses. Von nun an geht es um Künstliche Intelligenz. Schön gesagt, aber was ist Künstliche Intelligenz eigentlich – nun, ganz einfach, eine künstliche Form von normaler menschlicher Intelligenz!?! Hier wirft sich jedoch gleich die Frage auf, was Intelligenz eigentlich ist. Seit langem bereits streiten sich Mathematiker, Logiker und Philosophen darum, wie man Künstliche Intelligenz oder Intelligenz überhaupt definieren soll. Die Frage, ob der Mensch ein intelligentes Wesen ist, wird wohl jeder (Mensch zumindest) mit Ja beantworten. Es bietet sich daher an, einmal einzelne Handlungen eines Menschen zu betrachten, um dann zu begutachten, ob diese Handlungen als intelligent bezeichnet werden können oder nicht. Auf besonders interessante Aspekte in dieser Hinsicht werde ich im Kapitel »Irrgärten und Mäuse« zu sprechen kommen. Ich muß zugeben, daß ich mich bei der Zusammenstellung der Themen zum Kursabschnitt Künstliche Intelligenz wirklich sehr schwer tat. Wo fängt man am besten an? Wo hört man auf? Wo setzt man Schwerpunkte?

Als eine große Hilfe bei der Auswahl des Kursinhalts erwies sich mir das Buch »Gödel, Escher, Bach – ein endlos geflochtenes Band« von Douglas E. Hofstadter (ISBN 3-608-93039-X). Aufmerksam auf dieses Buch wurde ich durch den Logeleien-Kurs des 64'er, in dessen Rahmen dieses Buch empfohlen wurde. Es ist in der Tat eine Art von Bibel für alle KI-Freaks. Obgleich es keinerlei KI-Programme in Form von Listings zum Abtippen enthält, geht das Buch auf alle Bereiche der KI theoretisch in ausführlichem Maße ein. Das 800 Seiten starke Buch ist an vielen Stellen sehr kompliziert und unter Voraussetzung mathematischer Kenntnisse geschrieben. Trotzdem findet man in ihm zahlreiche Anregungen zum Schreiben von eigenen KI-Programmen. Je mehr ich mich mit KI und dem Schreiben dieses Kursabschnitts beschäftigte, umso mehr drängte sich in mir eine Frage auf: Die Frage, was uns KI eigentlich bringt – warum soll sich der Mensch den Kopf zerbrechen, um eine Maschine zu konstruieren, die so wie er (oder vielleicht noch besser?) denken und grübeln kann? Ich habe diese Frage bisher noch nicht beantworten können und glaube auch nicht, in allernächster Zukunft eine Antwort zu finden. Ich weiß jedoch, was uns KI auf dem C 64 im Rahmen dieses Kurses bringen soll: Nichts weiter als Spaß und Unterhaltung...

Bei KI-Programmierung auf dem C 64, oder sollte ich vielleicht besser sagen, Programmierung von simulierten menschlichen Denken, können in der Tat erstaunliche Resultate erzielt werden.

Besonders interessant ist dabei die Tatsache, daß zum Erzielen dieser Resultate oft sogar geringe Basic-Kenntnisse ausreichen. Die Programmierung von KI-Programmen erfordert ganz andere Überlegungen als die, die zum Schreiben von Spielen und Anwendungsprogrammen notwendig sind. Es geht einfach darum, simple logische Abläufe und Regeln in Programme umzusetzen. Das Problem besteht lediglich darin, die Logik in gewissen Abläufen zu erkennen. Auch hierzu werden im Kapitel »Irrgärten und Mäuse« wahre Musterbeispiele gegeben.

Bei der Einleitung zur KI möchte ich keine großen Worte über Grundlagen des logischen Denkens, logische Schlußfolgerungen oder Reproduzierbarkeit von Grundwissen verlieren. Über diese Themen wurden in letzter Zeit mehrere Artikel in den verschiedenen Fach- und Computerzeitschriften veröffentlicht. Es geht mir vielmehr darum, Ihnen anhand von praktischen Beispielen (Programmen) zu zeigen, wie KI-Simulationen funktionieren.

Am Ende dieses Kursteils können Sie mit Sicherheit selbst entscheiden, was Sie von KI halten sollen.
Genug der Einführung, jetzt geht's endlich los...

## Das NIM-Spiel

Jeder kennt es und hat es bestimmt auch schon einmal gespielt. Auf dem Tisch liegt eine Anzahl von Gegenständen (meist sind es Streichhölzer). Zwei Spieler dürfen nun abwechselnd 1, 2 oder 3 Gegenstände wegnehmen. Wer den letzten Gegenstand vom Tisch nehmen muß, hat schließlich verloren.

Listing 1 simuliert einen äußerst starken Spielpartner beim NIM-Spiel.

Sie werden schnell merken, wie spielstark dieses Programm ist. Zweifellos fraglich ist es allerdings, ob hier KI am

```
0 REM ****************************          <109>
1 REM *       KI-PROGRAMM 1        *          <081>
2 REM * 1985 BY MICHAEL NICKLES *            <157>
3 REM ****************************          <112>
10 REM DATA'S LESEN                           <162>
20 DIM X(20)                                  <088>
30 FOR I=1 TO 20                              <049>
40 READ X(I)                                  <253>
50 NEXT I                                     <134>
60 HZ=20: REM ANZAHL DER HOELZER              <034>
100 PRINT"{CLR}VOR IHNEN LIEGT EIN HAEUFCH
    EN MIT 20{4SPACE}HOELZERN."               <051>
110 PRINT"SIE UND DER COMPUTER KOENNEN NUN
    "                                         <112>
120 PRINT"ABWECHSELND JE 1, 2 ODER 3 HOELZ
    ER"                                       <130>
130 PRINT"WEGNEHMEN."                         <115>
140 PRINT"WER DAS LETZTE HOELZCHEN NEHMEN
    MUSS,"                                    <062>
150 PRINT"HAT VERLOREN."                      <038>
220 INPUT"MOECHTEN SIE ANFANGEN (J/N)";I$     <163>
230 IF I$<>"J"AND I$<>"N"THEN 220             <072>
240 IF I$="J"THEN PRINT"{DOWN}OK. SIE FANG
    EN AN.":GOTO 300                          <098>
250 IF I$="N"THEN PRINT"{DOWN}OK. DANN BEG
    INNE ICH.":GOTO 400                       <249>
300 :                                         <022>
301 REM SPIELER IST AM ZUG                    <036>
302 :                                         <024>
310 INPUT"WIEVIELE HOELZER";N                 <069>
320 IF N<>1 AND N<>2 AND N<>3 THEN PRINT"S
    IE KOENNEN NUR 1, 2 ODER 3 NEHMEN !":G
    OTO 310                                   <082>
330 HZ=HZ-N                                   <044>
340 IF HZ<=0 THEN PRINT"SIE HABEN VERLOREN
     !":END                                   <069>
400 :                                         <122>
401 REM COMPUTER IST AM ZUG                   <141>
402 :                                         <124>
410 N=X(HZ)                                   <092>
420 PRINT"ICH NEHME";N;"HOELZER."             <083>
430 HZ=HZ-N                                   <144>
440 IF HZ=0 THEN PRINT"SIE HABEN GEWONNEN
    !":END                                    <175>
500 :                                         <222>
501 REM HOELZERANZAHL FESTLEGEN               <164>
502 :                                         <224>
510 PRINT"ES LIEGEN JETZT NOCH";HZ;"HOELZE
    R DA !"                                   <165>
520 GOTO 300                                  <212>
1000 DATA 1,1,2,3,1,1,2,3,3,1,2,3,2,1,2,3,
     1,1,2,3                                  <151>
```

**Listing 1. Ein starker Gegenspieler im NIM-Spiel**

Werk ist. Stellen wir uns doch einmal zwei Kinder vor, die zum ersten Mal das NIM-Spiel mit Streichhölzern spielen. Die ersten Spiele werden so verlaufen, daß die Kinder abwechselnd 1, 2 oder 3 Hölzer wegnehmen, ohne groß nachzudenken, bis schließlich eines das letzte Hölzchen nehmen muß und als Verlierer dasteht. Wir nehmen einmal an, es liegen noch 4 Hölzer auf dem Tisch und Kind A ist dran. Ohne große Überlegung weiß es, daß es genau 3 Hölzer – nicht mehr (darf es gemäß der Regel ja auch gar nicht) und nicht weniger – wegnehmen muß, um Kind B das letzte Holz aufzuzwingen und so zu gewinnen. Würde Kind A nur 2 Hölzer wegnehmen, so würden auf dem Tisch 2 Hölzer liegen bleiben. Kind A weiß in diesem Fall, daß Kind B sodann 1 Holz nehmen und dadurch gewinnen würde. Es ist also mehr eine Frage der Zeit, bis die Kinder folgende Regel beziehungsweise Logik des NIM-Spiels durchschauen:

Regel: Liegen nur noch 4 Hölzer auf dem Tisch, so ist derjenige der Sieger, der bei dieser Stellung am Zug ist. Dies setzt natürlich voraus, daß der Ziehende logisch vorgeht – also 3 Hölzer wegnimmt und nicht dummerweise weniger und so den gewissen Sieg verschenkt!

Aufgrund dieser Regel wird es also jeder anstreben, am Zug zu sein, wenn sich 4 Hölzer auf dem Tisch befinden. Hingegen will niemand am Zug sein, wenn sich nur noch 5 Hölzer auf dem Tisch befinden, da er dann sicher verliert. Es wird früher oder später also jeder Spieler daraufhin arbeiten, daß der Gegner dann am Zug ist, wenn 5 Hölzer auf dem Tisch liegen. Und genau das tut das obige Programm. Voraussetzung ist jedoch, daß die Anzahl der Hölzchen bekannt ist, denn nur dann kann bis zum letzten Zug vorausberechnet werden. Bei unserem Programm sind 20 Hölzer festgelegt. Entsprechend sieht die Strategie des Programms aus, die im Feld X gespeichert ist.

Bild 1 zeigt die Vorgehensweise des Programms – wieviele Hölzer es nimmt, wenn eine bestimmte Anzahl von Hölzern auf dem Tisch liegt.



Bild 2 zeigt, auf welche 5 Punkte das Programm hinarbeitet, um den Spieler verlieren zu lassen. Aus Bild 2 geht auch hervor, daß der Spieler keine Gewinnchance hat, wenn er das Programm anfangen läßt. Liegen 20 Hölzer auf dem Tisch und das Programm beginnt, so nimmt es gleich 3 Hölzer weg. Der Spieler ist also erst beim ersten kritischen Punkt dran. Egal, was er macht, beim nachfolgenden Zug bringt ihn das Programm zum nächsten kritischen Punkt. Dies spielt sich solange ab, bis der Spieler beim letzten kritischen Punkt angelangt ist und somit verloren hat.

Es gibt für den Spieler also nur eine einzige Gewinnchance: Er muß dafür sorgen, daß das Programm immer bei den kritischen Punkten am Zug ist. Bei Bild 2 ist noch etwas

X(20)–X(1) →

| 3 | 2 | 1 | 1 | 3 | 2 | 1 | 2 | 3 | 2 | 1 | 3 | 3 | 2 | 1 | 1 | 3 | 2 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

Zug, wenn 20 Hölzer auf dem Tisch liegen.

Kritischer Punkt. Wer hier am Zug ist, verliert!

Zug, wenn 1 Holz auf dem Tisch liegt.

**Bild 1. Strategie des Spielprogramms**

| 3 | 2 | 1 | 1 | 3 | 2 | 1 | 2 | 3 | 2 | 1 | 3 | 3 | 2 | 1 | 1 | 3 | 2 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 1 | | | | 2 | | | | 3 | | | | 4 | | | | 5 |

**Bild 2. Fünf kritische Spielsituationen**

anderes auffällig: Alle kritischen Punkte haben von ihren benachbarten kritischen Punkten den gleichen Abstand! Nehmen wir einmal an, der Spieler beginnt und nimmt sofort 3 Hölzer weg. Das Programm landet so beim ersten kritischen Punkt und nimmt aus Verlegenheit ein Holz weg. Um das Programm zum nächsten kritischen Punkt zu bringen, muß der Spieler jetzt 3 Hölzer wegnehmen. Er muß also viel, viel Glück haben, wenn er gewinnen will.

Die Gewinn-Reihenfolge lautet: 3, 3, 2, 1, 3.

Dieses KI-Programm ist so primitiv, daß es auf jedem programmierbaren Taschenrechner, ja sogar mit drei Lämpchen, ein paar Drähten, Batterie und Schrauben verwirklicht werden kann (ist eine nette Geschenkidee!).

Ich hoffe, daß es mir anhand dieses Spiels gelungen ist, Ihnen schon etwas klarzumachen, wie man beim Programmieren von KI-Problemen vorgehen muß. Die Vorgehensweise sieht in etwa so aus:

1. Einfach mal mit dem Problem herumspielen, herumprobieren. Also ausprobieren: WAS PASSIERT WENN...
2. Ausprobieren, ob irgendwo logische Abläufe vorhanden sind (Bei unserem Beispiel: Logisch, daß man von 4 Hölzern auf dem Tisch 3 wegnehmen muß, wenn man am Zug ist und gewinnen will).
3. Versuchen, einfache Regeln aufzustellen.
4. Ein Gesamtkonzept entwickeln.
5. Skizzieren, Ablaufpläne erstellen.
6. Programmieren.

Sie sehen schon: Dem eigentlichen Programmieren der Lösung gehen zahlreiche Einzelschritte voran.

Die Gültigkeit dieser fünf Grundregeln kommt auch im folgenden Kapitel voll und ganz zur Geltung – beim Programmieren von Software-Mäusen, die sich in einem Labyrinth zurechtfinden lernen.

## Irrgärten und Mäuse – Grundlagen des »Backtracking«

Ein weiteres interessantes Gebiet, das uns noch tiefer in die Grundlagen der Künstlichen Intelligenz einführen wird, ist das Programmieren von »Mäusen«, die den Weg durch ein Labyrinth finden können – also KI-Programme, die Irrgärten »knacken« können. In den USA und in Japan werden zu diesem Thema regelrechte Wettkämpfe ausgetragen. Es geht darum, einen kleinen Roboter so zu konstruieren und programmieren, daß er möglichst schnell den Weg durch ein Labyrinth finden kann. Die Robotermäuse sind hierbei regelrechte Mini-Wunderwerke, zusammengebaut aus Motoren, Zahnrädern, Sensoren und natürlich vollgepackt mit Elektronik und einem Mikroprozessor. Wir wollen uns in diesem Kurs natürlich nicht mit Grundlagen der Mechanik zum Bau einer solchen Robotermaus vertraut machen (was unheimlich viel Geschick erfordert), sondern vielmehr ein Steuerprogramm für so eine Maus entwickeln.

Betrachten Sie bitte einmal das Labyrinth in Bild 3:

Als simpel oder primitiv kann man dieses Labyrinth mit Sicherheit nicht bewerten, denn man sieht auf den ersten Blick lediglich Eingang (Pfeil Ecke links unten) und Ausgang (Pfeil Ecke rechts oben), jedoch keine direkte Verbindung zwischen den beiden. Wir wollen an dieser Stelle einmal versuchen zu analysieren, wie ein Mensch vorgeht, wenn er den Weg durch dieses Labyrinth finden will. Zunächst einmal sieht man, wie bereits gesagt, auf einen Blick Eingang und Ausgang des Irrgartens. Kaum jemand wird jetzt versuchen, eine bestimmte Taktik des Vorgehens zu entwickeln – vielmehr tritt sofort das menschliche Auge in Kraft. Es versucht zunächst, eine möglichst direkte Verbindungslinie zwischen Eingang und Ausgang zu finden. In Sekundenschnelle stellt es dabei fest, daß ein direkter Weg nicht existiert – unter »direktem

Weg« verstehe ich einen Weg, der nur sehr wenige Verzweigungen hat. Aus menschlicher Sicht ist für diesen Vorgang kein lobenswerter Denkprozeß, keine Intelligenz notwendig. Der nächste Schritt ist schon etwas komplizierter. Wer zum allererstenmal mit einem solchen Irrgarten konfrontiert wird – zum Beispiel kleine Kinder – versucht nun vom Eingang ausgehend verschiedene Wege – meistens in Richtung Ausgang orientiert – anzusteuern. Man bedient sich hierbei Hilfsmittel wie dem Zeigefinger oder einem Stift zum Entlangfahren des gegangenen Weges.

Der versierte Irrgarten-Freak geht jedoch sicherlich anders vor. Aus seiner Erfahrung mit früheren Irrgärten hat er gelernt, daß Irrgartenkonstrukteure meistens einen sehr indirekten Weg vom Eingang zum Ausgang wählen (also möglichst viele Verzweigungen), um den Irrgarten möglichst kompliziert zu machen. Der Erfahrene wird mit Sicherheit auch nicht jeden erdenklichen Weg bis zum Ende ausprobieren. Er sieht gleich, wo Sackgassen-Gebiete sind, und geht erst gar nicht in diese hinein. Ein Sackgassengebiet ist ein Gebiet des Irrgartens, wo sich ein Weg in verschiedene Richtungen aufteilt, die jedoch alle in Sackgassen enden. Ein ganz deutliches Sackgassengebiet befindet sich in unserem Labyrinth in der linken unteren Ecke, gleich neben dem Eingang.

Sicherlich haben Sie an dieser Stelle bereits bemerkt, daß ich auf die Irrgärten und Mäuse besonders genau eingehen möchte. Dies hat seinen Grund, denn an diesem Beispiel läßt sich sehr gut demonstrieren, wie man vorgehen muß, wenn man intelligente Abläufe programmieren will.

Beim Programmieren von Action-Spielen und auch Adventures muß man sich oft keine allzu großen Grundgedanken machen. Natürlich ist es oft schwer, Ideen programmtechnisch umzusetzen (sei es das Programmieren von schnell bewegter Grafik oder auch Parser für Adventures), man hat jedoch gewisse Regeln, nach denen man sich richten oder an denen man festhalten kann. Bei KI-Programmierung ist dies anders: Die KI steckt noch in ihren Kinderschuhen und wird es höchstwahrscheinlich auch bleiben, solange der Mensch nicht eine Frage geklärt hat. Die Frage, was Intelligenz eigentlich ist, wann ein bestimmer Prozeß als intelligent bezeichnet werden kann und wann nicht. Man denke hierbei zum Beispiel an jemanden, der sich mit Hintergedanken, also mit Absicht dumm stellt, weil er glaubt, damit etwas besser erreichen zu können. Man stellt sich zum Beispiel dumm, weil man zu faul ist, etwas lang und breit zu erklären (zum Beispiel Grundlagen der KI....). Was ich sagen will ist, daß sich aus dem Verhalten eines Menschen heraus nie beurteilen läßt, ob eine Tat nun auf mangelnder Intelligenz oder auf bewußter Simulation von mangelnder Intelligenz beruht. Ich werde mich deshalb auch hüten, an dieser Stelle zu erörtern, wieviel beziehungsweise ob überhaupt Intelligenz zum Knacken von Labyrinthen erforderlich ist. Wir wollen deshalb nun vielmehr versuchen, herauszufinden, wie man ein Programm, das sich in Irrgärten zurechtfindet, programmiert und danach Analogien zum menschlichen Vorgehen suchen.

## Ein Profi im Labyrinth

Wie wir festgestellt haben, beruht die menschliche Vorgehensweise hauptsächlich auf visueller Basis, dem Durchchecken des Irrgartens mittels des Auges. Dies ist einfach, da wir das gesamte Labyrinth mit allen Wegen auf einmal überblicken können. Was wird jedoch, wenn wir es mit einem riesigen Labyrinth zu tun bekommen – zum Beispiel auf einer mehrere Quadratmeter umfassenden Papierfläche? Hier ist unsere »Optik« schnell überlastet. Völlig unbrauchbar wird die visuelle Methode schließlich, wenn wir uns vorstellen, daß es sich bei dem Irrgarten um einen unterirdischen, unbeleuchteten Komplex handelt, in dem wir uns direkt befinden. Um dann den Ausgang zu finden, braucht man eine gute Orientierungsfähigkeit (Intelligenz???) und eine große Portion Glück.

In dieser Situation können wir uns eine Robotermaus vorstellen. Sie befindet sich direkt im Irrgarten, hat keinen Gesamtüberblick, ja, weiß nicht einmal, in welcher Richtung sich der Ausgang befindet.

Was die visuelle Basis anbelangt, so wird der Mensch einem Homecomputer mit 64 KByte wie dem C 64 immer überlegen sein. Wenn es jedoch um Orientierungsfähigkeit im Dunkeln geht wie zum Beispiel in Irrgärten, die man nicht auf einmal überblicken kann (weil man sich zum Beispiel selbst in ihnen befindet), ist uns unser C 64 bei geschickter Programmierung bei weitem überlegen, da er Tausende von Schritten speichern kann, also eine Karte vom Irrgarten in seinem »Gedächtnis« mitschreibt. Der Mensch hingegen verliert je nach persönlicher Fähigkeit bereits nach (im Verhältnis zum Computer betrachtet) wenigen Abzweigungen die Orientierung.

Oder vielleicht doch nicht?

## Immer an der Wand lang

Im folgenden wollen wir ein einfaches Labyrinth betrachten – die Theorie läßt sich jedoch auch auf jegliche anderen Labyrinthe übertragen.

Das folgende »Schema« zeigt, wie man auch im Dunkeln, allein durch Tastsinn, todsicher aus jedem auch noch so komplexen Labyrinth herausfinden kann. Bei allen Experimenten setzen wir selbstverständlich logisch aufgebaute Labyrinthe voraus, das heißt, geht man von Ort A in nördliche Richtung, um zu Punkt B zu gelangen, so gelangt man von B zurück zu A, indem man sich nach Süden bewegt. Bei Labyrinthen in Adventures (besonders ZORK I) ist dies oft nicht der Fall!

Das Prinzip ist ganz einfach. Angenommen, wir betreten einen stockdunklen Irrgarten. Wir berühren sofort beim Eintreten mit der linken Hand die linke Wand und lassen diesen Kontakt für die gesamte Zeit, die wir uns im Labyrinth aufhalten, nicht abbrechen. Das heißt, wir laufen immer vorwärts, also nie zurück in die Richtung, aus der wir kommen und schleifen beim Vorwärtsgehen unsere linke Hand die Wand

**Bild 3. Ein Irrgarten für Mäuse**

entlang. Vorwärtsgehen kann natürlich auch um die Ecke gehen bedeuten. Wichtig ist nur, daß die linke Hand nie den Kontakt mit der Wand verliert.

Wenn Sie das Bild 4 betrachten und einmal mit Ihrem linken Zeigefinger (stellvertretend für die linke Hand) die rote Linie (in Pfeilrichtung) vom Eingang zum Ausgang entlangfahren, werden Sie das Prinzip verstehen.

Hat man erstmal durchgeblickt, ist die Methode sonnenklar und absolut logisch, ja, sogar primitiv – aber draufkommen muß man halt erst einmal. Trotzdem ist es fantastisch, wie einfach man einen irrsinnig komplexen Irrgarten unter Verwendung einer solch einfachen Regel knacken kann. Wer noch mißtrauisch ist (und das sollten Sie bei KI-Programmen immer sein!), kann unser Lösungsprinzip einmal auf den großen Irrgarten (Bild 1) anwenden – es funktioniert garantiert! Wie jede Medaille ihre Kehrseite hat, so hat jedoch auch unsere Methode ihre Haken. Sie ist zum Beispiel nur auf zweidimensionale Irrgärten anwendbar, das heißt, bei Irrgärten, in denen es auch auf und ab geht (zum Beispiel mit Treppen und Leitern), sind wir aufgeschmissen. Auch bei unlogischen Irrgärten, bei denen Richtungsänderungen irreversibel sind, kommen wir nicht sehr weit. Sie können nun einwerfen, daß es beim Irrgarten (Bild 4) viel schneller gehen würde, wenn man sich an der rechten Wand (Hand) orientieren würde. Das stimmt, aber auch hier haben Sie wieder Gebrauch von der visuellen Methode gemacht – und die gilt nicht. Im Prinzip ist es jedoch vollkommen egal, ob wir uns für rechts oder links entscheiden. Wichtig ist nur, daß man seine Entscheidung beim Laufen im Irrgarten immer beibehält. Interessant ist auch die folgende Tatsache: Linker und rechter Lösungsweg schneiden sich nie! – Ausprobieren!

Allerdings hat unsere Methode auch noch einen anderen Nachteil – man lernt nichts daraus. Das heißt, wenn man den Irrgarten ein zweites, drittes oder gar noch öfteres Mal durchläuft, braucht man immer die gleiche Zeit dazu (vorausgesetzt, man bewegt sich immer mit der gleichen Geschwindigkeit vorwärts). Günstiger wäre es, wenn man sich alle Sackgassen im Gedächtnis speichern würde und bei späteren Versuchen erst gar nicht mehr in diese hineinlaufen müßte – dieses »Speichern« ist bei unserer jetzigen Methode nicht möglich. Probieren Sie doch einmal beide Methoden im Irrgarten (Bild 1) aus, die links- oder die rechtsorientierte und die visuelle (die Speichermethode). Sie werden schnell merken, wie Sie bei mehrmaligem Durchwandern des Labyrinths mit der visuellen Methode immer mehr von Ihrem Erfahrungs-Speicher (Intelligenz???) Gebrauch machen werden. Mit der Zeit weiß man halt einfach immer besser, wann etwa man sich links orientieren muß, wann in Richtung Norden, wo man schon einmal eine falsche Richtung eingeschlagen hat (und dies deshalb nun meidet), und findet sich so immer besser zurecht. Nach einiger Zeit kennt man sich schließlich aus – man gelangt, ohne sich zu verirren, vom Eingang zum Ausgang.

**Bild 4. Die linke Hand an der Wand**

Schön und gut, Sie wissen jetzt eine Formel, mit der man sicher aus jedem zweidimensionalen, logisch aufgebauten Labyrinth herausfindet. Wie setzt man eine Lösungsmethode dieser Art jedoch zum Programm um?

Ich möchte Ihnen im folgenden ein Programm vorstellen, das folgende Kriterien erfüllt:

- Der Irrgarten muß nicht im Speicher des C 64 stehen. Das Programm ist in der Lage, jeden zweidimensionalen Irrgarten (mögliche Richtungen: N, S, O, W) mit beliebiger Größe, der in einer Form wie Bild 1 vorliegt, zu knacken.
- Das Programm fragt den »Spieler«, ob bestimmte Richtungen möglich sind. Es erwartet sodann immer eine Antwort in der Form JA oder NEIN. Man muß dem Programm mitteilen, wenn es den Ausgang gefunden hat.
- Das Programm bemerkt es, wenn man versucht, es mit falschen Antworten oder Irrgärten ohne Ausweg zu »leimen«.
- Das Programm merkt sich den richtigen Weg und findet beim zweiten Anlauf den Weg ohne jegliches Verirren.

## Rekursive Unterprogramme

Gerade das letzte Kriterium macht uns schwer zu schaffen, da es unser bisheriges Lösungskonzept völlig über den Haufen wirft.

Es muß also ein Neues her (das vom Aufbauprinzip her jedoch gar nicht so verschieden von unserer ersten Methode sein wird).

Die meisten Maus-Programme basieren auf einem Konzept, das die folgende Programmiertechnik erforderlich macht:

Rekursive Unterprogramme sind Unterprogramme oder auch Prozeduren, die sich selbst aufrufen. In höheren Programmiersprachen wie zum Beispiel LISP lassen sich rekursive Vorgänge relativ leicht programmieren. Basic ist hierzu nicht sehr geeignet. Das folgende Beispiel demonstriert auf einfache Art und Weise, wie rekursive Unterprogramme funktionieren:

```
10 REM REKURSIVES UP - DEMO
20 GOSUB1000:REM REKURSIVES UP AUFRUFEN
1000 REM REKURSIVES UNTERPROGRAMM
1010 :
1020 GOSUB1000:REM SELBSTAUFRUF
1030 :
1040 RETURN
```

Die Dokumentation hierzu ist kurz und einfach:

In Zeile 10 wird ein Unterprogramm aufgerufen, das in den Zeilen 1000 – 1040 steht. In Zeile 1020 – also innerhalb des Unterprogrammes – ruft es sich selbst auf.

Was wird wohl passieren, wenn wir dieses kurze Programm starten (was Sie ja eh schon getan haben, ohne auf meine Erklärung zu warten)?

Es kommt zu einem

OUT OF MEMORY ERROR IN 1020.

Mißrauisch, wie wir sind, geben wir nun

PRINT FRE(0)+65000 ein.

Dabei kommt heraus, daß 38323 Byte RAM frei sind. Wie kommt es denn dann zum OUT OF MEMORY ERROR, werden Sie sich nun sicher fragen.

Ganz einfach. Der C 64 kann nur eine begrenzte Anzahl von Unterprogrammen nach dem Stapelprinzip (als letztes rein, als erstes raus) verwalten und aufrufen.

Ändern Sie bitte Zeile 101 so ab:

1010 I=I+1.

Wenn Sie nun das Programm starten und nach seinem Abbruch den Wert I im Direktmodus abfragen, erhalten Sie den Wert 23. Das heißt, der C 64 ist in der Lage, maximal 23 Unterprogrammaufrufe zu speichern (die Rücksprungadressen). In anderen Worten läßt sich das Stapel-Prinzip so ausdrücken: Der C 64 findet eine GOSUB-Anweisung. Er nimmt sich nun einen Notizzettel und schreibt auf diesen die Zeile, in der die GOSUB-Anweisung erfolgt ist. Er merkt sich also, wo er weitermachen muß, wenn das Unterprogramm mit RETURN beendet wird. In dem eben notierten Unterprogramm kommt nun plötzlich wieder eine GOSUB-Anweisung vor. Es ist dabei völlig egal, ob ein anderes Unterprogramm aufgerufen wird oder ob sich das Unterprogramm rekursiv selbst aufruft wie in unserem Beispiel. Wieder nimmt der C 64 sich einen Zettel und notiert, wo es weitergeht, wenn dieses Unterprogramm mit RETURN beendet wird. Diesen Notizzettel legt der C 64 nun oben auf den ersten Notizzettel. Jetzt taucht im Programm eine RETURN-Anweisung auf. Woher soll der C 64 nun wissen, wohin er springen muß? Nun, er nimmt einfach den obersten Zettel vom Notizstapel weg (daher der Name Stapel-Prinzip). Auf diesem Zettel steht, wohin er zurückspringen muß. Nach dem Sprung wirft der C 64 den vom Stapel genommenen Notizzettel in den Papierkorb. Bei der nachfolgenden RETURN-Anweisung wird der nächste oberste Zettel vom Stapel genommen, dann der nächste und so weiter. Wenn eine RETURN-Anweisung kommt und der C 64 findet auf seinem »Schreibtisch« keinen Notizzettel, so meldet er sich empört mit RETURN WITHOUT GOSUB ERROR. Versucht man hingegen, ein 24. Unterprogramm aufzurufen, so meldet er sich mit OUT OF MEMORY ERROR IN. Auf seinem Schreibtisch ist leider nur Platz für maximal 23 Notizzettel. Der OUT OF MEMORY ERROR heißt praktisch nichts anderes als »Ich habe keinen Platz mehr für weitere Zettel auf meinem Schreibtisch«.

Der rekursiven Programmierung mittels GOSUB sind also enge Grenzen gesetzt.

Im Listing 2 finden Sie ein Maus-Programm, das auf solch rekursiven Unterprogrammen beruht:

Diese Maus funktioniert auf der Basis unserer »Immer-an-der-Wand-lang«-Methode«. Der Programmablauf ist äußerst einfach: Die Maus fragt einfach, ob sie nach links gehen kann. Ist dies der Fall, so speichert sie den ersten Zug und fragt, ob der Ausgang gefunden ist. Ist links nicht möglich, so versucht die Maus geradeaus zu gehen. Geht dies auch nicht, so versucht sie es mit rechts. Funktionieren alle drei Richtungen nicht (zum Beispiel, wenn wir die Maus beschummeln), stellt sie dies in Zeile 1110 fest und gibt die Suche auf. Die merkwürdige Abfrage mag zunächst verblüffend sein. Um die Maus zum Beispiel durch den Irrgarten in Bild 1 zu schicken, ist folgende Befehlsfolge notwendig:
geradeaus-rechts-geradeaus-links-rechts-rechts-links etc.

Bei der Eingabe ist zu beachten, daß sich Links-Rechts-Angaben nicht auf ihre Position (Spieler), sondern aus der Sicht der Maus und ihrer momentanen Bewegungsrichtung bezogen werden muß. Unsere Maus ist allerdings zu schwach für den Irrgarten in Bild 1. Irrgarten 1 erfordert nämlich 22 Richtungsänderungen bis zum Ausgang. Aufgrund der rekursiven GOSUB-Programmierung kann die Maus jedoch nur maximal 20 Schritte speichern (eigentlich 24, aber man soll sich immer etwas »Spielraum« lassen – deshalb Einschränkung auf 20).

```
1 REM ********************************       <051>
2 REM *   IRRGAERTEN UND MAEUSE I      *       <224>
3 REM * <C> 1985 BEI MICHAEL NICKLES *       <157>
4 REM ********************************       <054>
5 POKE 53280,0:POKE 53281,0:PRINT"{CLR,RVS
  ON,WHITE}MAUS I{RVOFF,DOWN}"               <026>
10 RI$(1)="LINKS"                            <103>
11 RI$(2)="GERADEAUS"                        <139>
12 RI$(3)="RECHTS"                           <204>
20 DIM M(20)                                 <000>
100 REM HAUPTPROGRAMM ----------------       <137>
101 :                                        <077>
110 GOSUB 1000                               <066>
120 PRINT"ICH FINDE KEINEN AUSGANG!":END     <035>
1000 REM REKURSIVES UP ---------------       <001>
1001 :                                       <215>
1005 I=0                                     <176>
1010 I=I+1                                   <231>
1015 PRINT"KANN ICH ";RI$(I); " GEHEN (J/N
     )"                                      <011>
1020 INPUT BE$                               <002>
1025 IF LEFT$(BE$,1)="N"THEN 1110            <143>
1030 M(V)=I                                  <094>
1050 PRINT"AUSGANG GEFUNDEN (J/N)"           <180>
1055 INPUT BE$                               <039>
1060 IF LEFT$(BE$,1)="J"THEN PRINT"GESCHAF
     FT !":END                               <130>
1065 IF V=20 THEN PRINT"HABE MICH VERIRRT
     !":END                                  <132>
1070 V=V+1:GOSUB 1000                        <122>
1095 PRINT"AUSGANG GEFUNDEN (J/N)"           <225>
1100 V=M(T)                                  <196>
1110 IF I=3 THEN RETURN                      <151>
1120 GOTO 1010                               <076>
```

**Listing 2. Eine »dumme« Maus**

## Grundlagen des Backtracking

Wer glaubt, daß es auch ganz ohne Rekursives geht, hat recht:

Zu dem Programm in Listing 3 habe ich nichts zu sagen. Sie sehen es sicher selbst. Aus der Links-Rechts-Methode läßt sich einfach nichts herausholen, was auch nur im entferntesten mit KI zu tun hat.

Backtracking – diese Bezeichnung trifft bei Programmen zu, die verschiedene Lösungen ausprobieren und einen Lösungsweg, der nicht zum Ziel führt, »rückwärts« wieder verlassen können, um eine andere Lösungsalternative auszuprobieren.

Unsere nächste Maus soll nach solch einem Backtracking-Verfahren arbeiten. Machen wir uns einmal ein paar Gedanken über unser neues Programm.

Nehmen wir einmal an, die Maus arbeitet ähnlich wie wir, wenn wir ein Adventure spielen. Sie zeichnet sich eine Karte des Irrgartens. Bei jedem Schritt, den die Maus tiefer in das Labyrinth vordringt, wird diese Karte größer. Anhand der Karte kann die Maus jederzeit feststellen, wo sie schon war, welchen Weg sie bisher gegangen ist.

Hier noch einmal eine kleine Definition in bezug auf zweidimensionale, logische Labyrinthe:

```
1 REM ********************************       <051>
2 REM *   IRRGAERTEN UND MAEUSE II     *       <083>
3 REM * <C> 1985 BEI MICHAEL NICKLES *       <157>
4 REM ********************************       <054>
5 POKE 53280,0:POKE 53281,0:PRINT"{CLR,RVS
  ON,WHITE}MAUS II{RVOFF,DOWN}"              <188>
10 RI$(1)="LINKS"                            <103>
11 RI$(2)="GERADEAUS"                        <139>
12 RI$(3)="RECHTS"                           <204>
100 REM HAUPTPROGRAMM ----------------       <137>
110 I=0                                      <043>
120 I=I+1                                    <103>
125 IF I=4 THEN PRINT"HABE MICH VERIRRT !"
    :END                                     <156>
130 PRINT"KANN ICH ";RI$(I);" GEHEN (J/N)"   <144>
140 INPUT BE$                                <140>
150 IF BE$="N"THEN 120                       <077>
155 I=0                                      <088>
160 PRINT"AUSGANG GEFUNDEN (J/N)"            <050>
170 INPUT BE$                                <170>
180 IF BE$="J"THEN PRINT"PRIMA !":END        <252>
200 GOTO 120                                 <160>
```

**Listing 3. Das Maus-Programm ganz einfach**

Bei jedem Irrgarten kann man den Ausgang mit dem Eingang durch den absoluten Lösungsweg verbinden. Der Lösungsweg ist also der einzige, kürzeste Weg, der notwendig ist, um vom Eingang zum Ausgang zu gelangen.

Alle anderen Wege (eventuelle Verzweigungen des Lösungswegs) enden mehr oder weniger in Sackgassen. Unsere Maus betritt den Eingang und geht bis zur ersten »Kreuzung«. Der Weg vom Eingang bis zur ersten Kreuzung ist mit absoluter Sicherheit ein Teilstück des Lösungsweges. Kommt die Maus an der ersten Kreuzung an, gibt es folglich nur zwei Möglichkeiten:

a. Sie schlägt einen neuen Weg ein, der wiederum einen Abschnitt des Lösungsweges repräsentiert. Sie bleibt also auf dem richtigen Weg.
b. Die Maus kommt vom einzig richtigen Lösungsweg ab und gerät in eine Sackgasse oder sogar in ein Sackgassengebiet. Dieser Fall tritt mit Sicherheit häufiger als Fall a ein.

Wir wollen der Einfachheit halber zunächst einmal folgendes festlegen: Jeder Weg, der vom Lösungsweg wegführt, endet früher oder später in einer absoluten Sackgasse, aus der nur ein einziger Weg hinausführt: der Weg, durch den man in sie geraten ist (zum Beispiel ein langer Korridor, der plötzlich endet).

Unsere Maus erhält von uns einen Radiergummi. Mit ihm kann sie alle Wege, die sie bereits in ihre Karte gezeichnet hat und die als absolute Sackgasse enden, wieder herausradieren. Die Karte, die dann letztendlich übrigbleibt, stellt nur noch den absoluten Lösungsweg dar. Dieses Herausradieren, also das Widerrufen bereits gefällter Entscheidungen, die sich als falsch erwiesen haben – das Löschen falscher oder unbrauchbarer Notizen – bezeichnet man als den eigentlichen Backtracking-Vorgang. Das folgende Schema zeigt den Irrgarten, bei dem alle falschen Wege wegradiert wurden. Der Weg, der noch vorhanden ist, ist der absolute Lösungsweg (Bild 5).

Bild 5 zeigt den Irrgarten von Bild 4, nachdem er von unserer nächsten Maus durchwandert worden ist.

Das heißt, alles, was vom Irrgarten noch übriggeblieben ist, ist der absolute Lösungsweg (auch durch Pfeil deutlich gemacht). Der restliche Teil des Irrgartens ist entfernt worden. Die schwarz ausgemalten Wege sind von der Maus als absolute Sackgassen analysiert worden. Nachdem die Maus einen solchen Weg durch die einzig mögliche Richtung wieder verlassen hat, entfernt sie ihn sofort für immer aus ihrem Gedächtnis. Im Labyrinth sind jedoch noch zwei andere ausgemalte Flächen zu sehen: eine gelbe und eine rote. Für die folgenden Überlegungen wollen wir uns auf die Wegabschnitte X, A, B, C und D beziehen.



**Bild 5. Nur der Lösungsweg bleibt übrig**

Nehmen wir einmal an, unsere Maus steht an der Stelle X. Sie befindet sich also im rot ausgemalten Wegteil. Wenn wir einmal von der Richtung absehen, aus der sie gerade gekommen ist, hat die Maus nun drei Möglichkeiten. Sie kann zu A, B oder C gehen. Unsere Maus entscheidet sich zunächst für A. Leider erweist sich A als absolute Sackgasse. Die Maus geht also zurück zu X und vergißt A (wird schwarz ausgemalt). Nun versucht die Maus es bei B. Wieder Pech. Also zurück zu X und B vergessen. Nun geht die Maus nach C. Sie weiß an dieser Stelle noch nicht, daß auch dieser Wegabschnitt nach der nächsten Abbiegung in einer Sackgasse enden wird. Von C aus hat die Maus zwei Möglichkeiten: Sie kann nach X und nach D gehen. Unsere Maus ist jedoch so programmiert, daß sie nur und wirklich nur dann zurück zu dem Punkt geht, aus dem sie gerade kommt, wenn dies die einzige Möglichkeit ist. Sie hat also das Bestreben, immer tiefer in das Labyrinth vorzudringen. Sie zieht stets die unbekannten Wegabschnitte den bereits begangenen vor.

## Die Programmierung der dritten Maus

Diese Regel zwingt die Maus also nun dazu, nach Punkt D zu gehen. In Punkt D stellt sie jedoch wieder fest, daß sie in einer absoluten Sackgasse gelandet ist. Sie geht somit die einzig mögliche Richtung zurück (zurück zu C) und vergißt Wegabschnitt D. Durch das Eliminieren des Wegteils D ist nun auch C zur absoluten Sackgasse geworden. Die Maus wählt wieder die einzig mögliche Richtung, geht nach X und eliminiert Wegteil C. Dadurch, daß die Wegteile A, B, C eliminiert worden sind, ist nun auch Teil X zur absoluten Sackgasse geworden. Die Maus nimmt also die einzige Richtung ein, die von X aus noch möglich ist. Sie landet dadurch wieder auf dem richtigen Lösungsweg. Sie sehen also, wie unsere Maus arbeitet, wenn sie in einer Sackgasse gelandet ist. Aber wie setzt man dieses Wissen über Mäuse in ein Programm um? Wie uns bereits klar ist, braucht unsere neue Maus ein Gedächtnis, mit dem sie sich merken kann, wo sie bereits war – somit also letztendlich den absoluten Lösungsweg. Dieses Gedächtnis muß flexibel sein. Das heißt, es muß sowohl speichern als auch vergessen können. Zunächst erweitert unsere Maus ihr Gedächtnis mit jedem Schritt vorwärts. Gerät sie jedoch in eine Sackgasse, so muß sie einen Schritt rückwärts machen und somit bereits Gelerntes, das sich als falsch, nämlich als Sackgasse erwiesen hat, wieder vergessen. Solch ein flexibles Gedächtnis wird in unseren Mausprogrammen stets mittels eines dimensionierten Feldes realisiert.

Wie muß dieses Feld aussehen?

Wir haben definiert, daß die Irrgärten, in denen sich unsere Maus zurechtfinden wird, zweidimensional logisch aufgebaut sein müssen. Es gibt in jedem Raum genau vier mögliche Richtungen: N, S, O und W. Im Prinzip könnten wir auch Richtungen wie NW, SW, NO und SO erlauben. Wichtig ist nur, daß alle Richtungen in der gleichen Ebene verlaufen. Der Einfachheit halber beschränken wir uns jedoch auf die vier Grundhimmelsrichtungen. Was liegt nun näher, als eine Tabelle anzulegen, die zu jedem Raum die möglichen Richtungen speichert.

Dieses Gedächtnis sieht dann aus wie Tabelle 1.

Die Tabelle 1 ist wie folgt zu interpretieren:

Wenn man von Raum 0 aus nach Norden geht, gelangt man zu Raum 1. Geht man nach Osten, so gelangt man zu Raum 2 und so weiter.

Hoppla, ist das nicht etwa eine Landkarte, wie man sie zur Programmierung von Abenteuerspielen benötigt?

Richtig. Dies ist genau die Methode, die ich im ersten Adventure-Programmierkurs zum Definieren der einzelnen

Räume eines Spiels festgelegt habe (wer will, kann sie in Sonderheft 2/85 nachlesen!).

Für das »Gedächtnis« unserer Maus legen wir folgendes fest:

1. Zu Beginn des Programmes erhält jedes Element der Tabelle den Wert 1. Das heißt, jede Möglichkeit, die die Maus noch nicht ausprobiert hat, hat den Wert 1.
2. Will die Maus in eine Richtung gehen, die nicht möglich ist, so erhält die entsprechende Stelle den Wert 2. Sackgassen kennzeichnen sich also durch dreimaliges Vorkommen einer 2 in ihrer »Zeile«.
3. Ist eine Richtung möglich, so wird in die entsprechende Tabellenstelle die Nummer des Raumes geschrieben, zu dem diese Richtung führt.

Je größer die Tabelle ist, umso größer ist die Gedächtnisleistung der Maus. 38 KByte Basic-RAM sind genug, um der Maus ein Gedächtnis für über 1000 Räume zur Verfügung zu stellen.

Natürlich benötigen wir auch noch einen Zeiger, der immer auf den Raum der Tabelle zeigt, in dem sich die Maus gerade befindet. Vielleicht finden Sie es noch verwirrend, wie man bei einem Labyrinth überhaupt von Räumen sprechen kann, beziehungsweise, was die Zeiger 1 und 2 sollen. Die folgenden Beispiele werden es Ihnen klarmachen:

In Bild 6 sind die ersten drei Schritte der Maus eingezeichnet. Die Maus beginnt ihre Suche immer in Raum 0, dem Labyrintheingang oder dem Ausgangspunkt, wenn die Maus inmitten eines Labyrinthes ausgesetzt wird. Die Maus läuft nun nach Norden zu Raum 1. Ein Raum kann also als Wegkreuzung angesehen werden. Wenn uns die Maus später fragt, ob sie nach Norden gehen kann, will sie nicht wissen, wie weit sie nach Norden gehen kann, sondern lediglich, ob sie bis zur nächsten Kreuzung in nördlicher Richtung gehen kann (dies schließt natürlich auch Sackgassen ein). Von Punkt 1 aus bieten sich der Maus vier Richtungen an. Süden scheidet für sie zunächst aus, da sie, wie wir bereits festgelegt haben, immer unerforschte Gebiete bevorzugt. Die Maus geht nach Norden weiter. Warum eigentlich gerade nach Norden? Ganz einfach. Die Maus geht nach einem festgelegten Schema vor – N, O, S, W. Das heißt, sie versucht immer zuerst nach Norden zu gehen. Gelingt dies nicht, so versucht sie es mit Osten, dann mit Süden und mit Westen.

Entsprechend wird die endgültige Maustabelle auch definiert (Spartenreihenfolge: N-O-S-W). So gelangt die Maus schließlich zu Raum 2. Im Gedächtnis der Maus sieht es jetzt aus wie in Tabelle 2.

Bisher hat die Maus noch nie eine negative Antwort auf einen Richtungsversuch bekommen. Daher enthält die Tabelle 2 keine 2-Elemente: Woher weiß die Maus eigentlich, daß man von Raum 1 mittels »Süden« nach Raum 0 gelangt – sie hat dies doch noch gar nicht ausprobiert!!!

Unsere Maus ist nicht dumm. Sie erlaubt sich einen kleinen Kunstgriff, der sich so formulieren läßt:

**Regel 1**

Gelange ich mittels Befehl X (»N«) von Raum A zu Raum B, so kann ich bei logischen Systemen (Labyrinthen) voraussetzen, daß ich durch den Gegenbefehl zu X (»S«) von Raum B zu Raum A gelange.

Die Maus macht also nicht nur zu den aktuellen Vorgängen in ihr Gedächtnis Einträge, sondern gleichzeitig auch zu dem Gegenvorgang. Deshalb hat sie in der Spalte S bei Raum 1 sofort eine 0 eingetragen, nachdem sie in der N-Spalte von 0 eine 1 eingetragen hat. Der Zeiger des Gedächtnisses steht auf Raum 2 – dort, wo sich die Maus gerade befindet.

Jetzt wird's spannend. Die Maus ist in einer Sackgasse gelandet. Dies ist der Anfang vom Backtracking-Verfahren. Die Maus versucht vergeblich, eine andere Richtung als die, die sie einen Schritt zurückwirft, zu finden, denn sie will ja immer vorwärts. Enttäuscht muß sie ihr Gedächtnis mit 2 füllen. Dies verdeutlicht Tabelle 3.

Gemäß unserer Definition sind Räume mit drei 2-Elementen in einer Zeile absolute Sackgassen.

Die Maus weiß das auch und entschließt sich zu einem Rückwärtsschritt. Dazu setzt sie den Zeiger auf ihr Gedächtnis um 1 zurück. Raum 2 muß nun eliminiert (schwarz ausgemalt) werden. Das heißt, alle Wege, die zu Raum 2 führen, müssen zerstört werden. Natürlich gibt es nur einen Weg zu Raum 2, da nur ein Weg aus einer Sackgasse herausführen kann. Die Maus weiß auch ganz genau, welcher Weg dies ist. Wir definieren:

**Regel 2**

Wenn ein Versuchsergebnis falsch ist, dann muß ich einen Schritt zurückgehen (Backtracking). Durch den Befehl X bin ich von A nach B (in eine Sackgasse) geraten. Ich muß also zurück zu A. Dies erreiche ich durch den Gegenbefehl von X. Wenn ich wissen will, durch welchen Befehl (Richtung) ich von A nach B gelangt bin, so muß ich den Gegenbefehl zum Gegenbefehl von X bilden (Regel 3). Diesen muß ich dann aus meinem Gedächtnis löschen, um nicht noch einmal zu diesem negativen Ergebnis zu gelangen.

**Regel 3**

Wenn man vom Gegenbefehl eines Befehls X wiederum den Gegenbefehl bildet, so erhält man wieder den ursprünglichen Befehl X (Analogie: Bilden von Kehrwerten).

Anhand dieser drei Regeln läßt sich nahezu jeder Backtracking-Vorgang (»Aus Irrtum lernen«) beschreiben.

| RAUM | N | S | O | W |
|---|---|---|---|---|
| 0 | 1 | – | 2 | – |
| 1 | – | 0 | – | – |
| 2 | – | 3 | – | 0 |
| 3 | 2 | – | – | – |

Tabelle 1

| Zeiger auf Maus-Position | RAUM | N | O | S | W |
|---|---|---|---|---|---|
| | 0 | 1 | -1 | -1 | -1 |
| | 1 | 2 | -1 | 0 | -1 |
| → | 2 | | | 1 | |

Tabelle 2

| | RAUM | N | O | S | W |
|---|---|---|---|---|---|
| | 0 | 1 | -1 | -1 | -1 |
| | 1 | 2 | -1 | 0 | -1 |
| → | 2 | -2 | -2 | 1 | -2 |

Tabelle 3

| | RAUM | N | O | S | W |
|---|---|---|---|---|---|
| | 0 | 1 | -1 | -1 | -1 |
| | 1 | 2 | -1 | 0 | -1 |
| → | 2 | -2 | -2 | 1 | -2 |

Tabelle 4

| | RAUM | N | O | S | W |
|---|---|---|---|---|---|
| | 0 | 1 | -1 | 1 | -1 |
| → | 1 | -2 | -1 | 0 | -1 |

Tabelle 5

Der Lösungsweg:
N-O-N-O-S

| RAUM | N | O | S | W |
|---|---|---|---|---|
| 0 | (1) | | | |
| 1 | -2 | (2) | 0 | |
| 2 | (3) | | | 1 |
| 3 | -2 | (4) | 2 | |
| 4 | 5 | 5 | (5) | 3 |
| 5 | 2 | 2 | 4 | -2 |
| 5 | 6 | 6 | 6 | 4 |
| 6 | 2 | -2 | 5 | -2 |
| 6 | 2 | -2 | -2 | 5 |
| 6 | 5 | 7 | -2 | -2 |
| 7 | -2 | -2 | -2 | 6 |
| 5 | 4 | | | |

Tabelle 6

Aus Regel 2 weiß die Maus, daß sie den Wert 2 in Raum 2, Spalte N auf 2 setzen muß. (Tabelle 4)

Nach dem Backtracking sieht es im Gedächtnis aus wie in Tabelle 5. Die Maus steht jetzt wieder in Raum 1. Das neue Irrgartenbild entspricht Bild 7.

Wie macht die Maus nun weiter?

Nun, sie hat doch ihr N-O-S-W-Schema. Norden hat sie in Raum 1 schon ausprobiert. Also ist jetzt Osten dran. Stellt die Maus nach einem abgeschlossenen Backtracking fest, daß ihre jetzige Position wieder drei Zweien enthält, führt sie einen weiteren Backtracking aus (Vergleichen Sie das Beispiel in Bild 2, Raum X).

Sicher haben Sie bereits bemerkt, daß ich für die letzten Beispiele den Ausgang des Labyrinths als Eingang benutzt habe. Ich tat dies, um schneller in eine Sackgassensituation zu gelangen. Die Tabelle 6 zeigt, wie das Gedächtnis der Maus arbeitet, wenn die Maus vom Eingang links unten ausgehend (wie in Bild 1) ihren Weg zum Ausgang sucht (Bild 8). Das fertige Maus-Programm finden Sie in Listing 4.

**Dokumentation zum Listing 4**

| Variablen | M%: | Größe des Gedächtnisses |
|---|---|---|
| | M(M%,3): | Gedächtnis |
| | MZ: | Zeiger auf Gedächtnis |
| | X: | Befehl (N, S, O oder W) |
| | N = 0 | |
| | O = 1 | |
| | S = 2 | |
| | W = 3 | |

Die restlichen Variablen sind lediglich Hilfsvariablen.

Bild 6. Die ersten Schritte der Maus

Bild 7. Die Situation nach dem ersten Backtracking

**Unterprogramme:**

| | |
|---|---|
| 5000 bis 5150 | Unterprogramm für Backtracking-Vorgang. |
| 5000 bis 5050 | Feststellen, ob die Maus sich in einer Sackgasse befindet. Schleife durchläuft M(MZ,0)–M(MZ,3), also den Standort der Maus – und zählt, wie oft der Wert 2 vorkommt. |
| 5100 | Beginn des Backtrackings. |
| 5110 | Letzter Schritt, der in die Sackgasse führte, wird zurückgenommen – der Gedächtniszeiger wird um 1 erniedrigt. |
| 5120 bis 5140 | Der falsche Schritt wird aus der Tabelle entfernt und mit dem Wert 1 gefüllt. Alle Verbindungswege zur Sackgasse (beziehungsweise der einzige Weg) werden gelöscht – auf 2 gesetzt. Durch Bilden des Gegenbefehls zu X erkennt die Maus, wie sie aus der Sackgasse entkommt und meldet sich in Zeile 5137 zu Wort. |
| 6000 bis 6080 | Unterprogramm zum Bilden des Gegenbefehls. Aus Norden wird Süden, aus Osten Westen und so weiter. |
| 10000 bis10099 | Unterprogramm zur Ausgabe des bisherigen Weges. Hier wird in einer Schleife das Gedächtnis von Raum 0 - MZ 1 durchlaufen. In jedem Raum sind allerdings immer zwei Richtungen gespeichert. Beide führen zu Abschnitten des Lösungsweges: die eine in Richtung Ausgang, die andere in Richtung Eingang. Durch die IF-THEN-Abfrage in Zeile 10045 wird erreicht, daß nur der vorwärtsbringende Befehl (also Richtung Ausgang) ausgegeben wird. |
| 5 bis 30 | Vorbereitungen zum Programmablauf. |
| 40 bis 48 | Schleife zum Löschen des Gedächtnisses. |
| 100 bis 160 | Hier wird die Eingabe der Ausgangsrichtung gefordert (siehe Erklärung zum Ablauf). |
| 1000 bis 1210 | Steuerprogramm der Maus. |
| 1110 | X=0. Das heißt, die Maus versucht als erste Richtung Norden. Ihr Schema lautet schließlich N-O-S-W. X kann bis zum Wert 3 (=Westen) erhöht werden. So werden alle möglichen Richtungen nacheinander ausprobiert. |
| 1125 | Hier fragt die Maus, ob die Richtung X möglich ist. Vorher prüft sie jedoch in Zeile 1120, ob sie diese Richtung in diesem Raum nicht schon einmal ausprobiert hat. |

Bild 8. Der Weg der Maus vom Eingang zum Ausgang

Nur wenn an der entsprechenden Stelle im Gedächtnis eine 1 steht, versucht sie es mit X. Wenn X nicht geht, so wird eine 2 gemerkt. X wird um 1 erhöht und der nächste Versuch beginnt. Ist die Richtung möglich, so merkt sich die Maus diese (Zeile 1141) und bewegt sich in diese Richtung (Zeile 1142). In Zeile 1144 gibt die Maus auf, wenn ihr Gedächtnis voll ist – sie ist also schon M%-Schritte gegangen, die sich alle als richtig erwiesen haben.

1146 bis 1152 Hier wird der Gegenbefehl gebildet und im neuen Raum gespeichert. Dies ist nötig, um die Priorität von unerforschten Gebieten gegenüber bereits bekannten zu gewährleisten (Zeile 1210).

**Bedienungsanleitung zur dritten Maus**

Nach dem Programmstart müssen Sie sich ein wenig gedulden, da die Maus erst ihr Gedächtnis löschen muß.

Dann werden Sie nach der Anfangsrichtung gefragt. Es ist die Richtung, die von Raum 0 (Eingang) zu Raum 1 führt. Dies ist notwendig, damit die Maus den Eingang niemals als Sackgasse interpretiert.

Sodann stellt die Maus Fragen, zum Beispiel:

»Kann ich nach Norden gehen?« Sie müssen dann »J« oder »N« antworten. Ab und zu gibt die Maus auch automatisch an, wohin sie will (um aus einer Sackgasse rauszukommen). Sie sagt dann zum Beispiel »Ich gehe nach Norden«. Durch Drücken der SPACE-Taste gibt die Maus ihren Gedächtnisstand aus – also den Weg, den sie bisher vom Eingang aus gegangen ist.

Unsere dritte Maus ist schon ziemlich schlau. Sie bemerkt es, wenn man sie beschummelt und hat auch mit dem Irrgarten von Bild 1 keinerlei Probleme. Allerdings ist ihr Gedächtnis auf 1000 Richtungen begrenzt, die sie sich merken kann.

Mit Hilfe einer relativen Datei kann dieses Gedächtnis auf bis zu 15000 Räume vergrößert werden. Sie können so eine Änderung mit ihrem bisher erworbenen Wissen leicht selbst vornehmen.

```
1 REM ******************************      <051>
2 REM *   IRRGAERTEN UND MAEUSE III   *   <057>
3 REM * <C> 1985 BEI MICHAEL NICKLES *    <157>
4 REM ******************************      <054>
5 POKE 53280,0:POKE 53281,0:PRINT"{CLR,RVS
  ON,WHITE}MAUS III{RVOFF,DOWN}"          <254>
10 M%=1000:DIM M(M%,3):REM GEDAECHTNIS    <180>
20 RI$(0)="NORDEN"                        <126>
22 RI$(1)="OSTEN"                         <046>
24 RI$(2)="SUEDEN"                        <117>
26 RI$(3)="WESTEN"                        <042>
30 MZ=1            :REM GED. ZEIGER       <073>
40 FOR I1=0 TO M%    :REM GEDAECHTNIS     <118>
42 : FOR I2=0 TO 3   :REM LOESCHEN        <043>
44 :    M(I1,I2)=-1                       <081>
46 : NEXT I2                              <004>
48 NEXT I1                                <151>
100 PRINT"BIITE GEBEN SIE DIE ANFANGSRICHT
    UNG EIN:                              <115>
110 PRINT"NORDEN=1                        <198>
120 PRINT"OSTEN =2                        <148>
130 PRINT"SUEDEN=3                        <185>
140 PRINT"WESTEN=4                        <148>
150 GET I$:IF VAL(I$)=0 OR VAL(I$)>3 THEN
    150                                   <175>
160 X=VAL(I$)-1:GOTO 1141                 <197>
1000 REM HAUPTPROGRAMM -------------------
     ----------------------------------
     -                                    <044>
1001 :                                    <215>
1100 REM RICHTUNG AUSPROBIEREN            <242>
1101 :                                    <061>
1110 X=0             :REM 1. VERSUCH = N  <070>
1115 :  GOSUB 5000   :REM SACKGASSE ?     <249>
1120 :  IF M(MZ,X)<>-1 THEN 1200          <167>
1125 :  PRINT"KANN ICH NACH ";RI$(X);" GEH
     EN ?";             :REM X PROBIEREN  <240>
1130 :  GET I$:IF I$<>"J"AND I$<>"N"AND I$
     <>" "THEN 1130                       <232>
1131 :  IF I$=" "THEN 10000               <087>
1135 :  IF I$="N"THEN PRINT" NEIN.":M(MZ,X
     )=-2:X=X+1:GOTO 1115 :REM X GEHT NICH
     T                                    <002>
1140 PRINT" JA."      :REM X GEHT         <193>
1141 :  M(MZ,X)=MZ+1:REM RICHT. MERKEN    <188>
1142 :  MZ=MZ+1       :REM BEWEGEN        <008>
1144 :  IF MZ>M%THEN PRINT"HABE MICH VERIR
     RT - GEBE AUF !":END:REM MEMORY FULL <095>
1146 :    G=X:GOSUB 6000:X=G
                      :REM GEGENBEFEHL    <094>
1148 :    M(MZ,X)=MZ-1                    <241>
1150 :               :REM MERKEN          <104>
1152 :    G=X:GOSUB 6000:X=G
                      :REM GEGENBEFEHL    <100>
1160 :  GOTO 1000                         <001>
1200 :  IF M(MZ,X)=-2 THEN X=X+1:GOTO 1115
                          :REM X GEHT NIC
     HT                                   <234>
1210 :  IF M(MZ,X)>-1 THEN X=X+1:GOTO 1115
                          :REM PRIORITAET
      !!!                                 <011>
5000 REM FESTSTELLEN OB MAUS IN SACKGASSE
     IST (3 * -2) ----------------------
     -                                    <251>
5001 :                                    <151>
5010 IC=0                                 <040>
5015 FOR N=0 TO 3                         <040>
5020 : IF M(MZ,N)=-2 THEN IC=IC+1         <164>
5030 NEXT N                               <072>
5040 IF IC<3 THEN RETURN                  <037>
5050 IF IC=4 THEN PRINT"ICH LASSE MICH NIC
     HT REINLEGEN !":END                  <030>
5100 REM BACKTRACKING - SCHRITT ZURUECK --
     ----------------------------------
     -                                    <198>
5101 :                                    <251>
5110 MZ=MZ-1    :REM ZEIGER UM 1 ZURUECK  <070>
5120 FOR N=0 TO 3                         <147>
5125 : M(MZ+1,N)=-1
                 :REM IRRWEG LOESCHEN     <184>
5130 : IF M(MZ,N)<>MZ+1 THEN 5140         <173>
5134 : M(MZ,N)=-2
                 :REM IRRWEG LOESCHEN     <014>
5136 : G=N:GOSUB 6000                     <121>
5137 : PRINT"ICH GEHE NACH "RI$(G)"."     <139>
5140 NEXT N                               <184>
5145 X=0        :REM RESET AUF N          <165>
5147 IF MZ=1 THEN PRINT"BIN WIEDER AM EING
     ANG!":POKE 109,0:WAIT 198,1:RUN      <026>
5150 GOTO 5000  :REM PRUEFEN              <253>
6000 REM GEGENBEFEHL BILDEN -------------
     ----------------------------------
     -                                    <181>
6001 :                                    <135>
6010 : IF G=0 THEN G=2:GOTO 6080          <144>
6020 : IF G=1 THEN G=3:GOTO 6080          <222>
6030 : IF G=2 THEN G=0:GOTO 6080          <028>
6040 : IF G=3 THEN G=1:GOTO 6080          <106>
6080 RETURN                               <040>
10000 REM LOESUNG AUSGEBEN --------------
      ----------------------------------
      --                                  <218>
10001 :                                   <071>
10010 PRINT"{CLR}DURCH FOLGENDE BEFEHLSFOL
      GE GELANGT MAN VOM EINGANG ZU MEINEM
       ";                                 <229>
10020 PRINT"JETZIGEN STANDORT:"           <191>
10030 FOR I1=0 TO MZ-1                    <203>
10040 : FOR I2=0 TO 3                     <034>
10045 :    IF M(I1,I2)>0 AND M(I1,I2)<>I1-1
       THEN PRINT RI$(I2)",";:I2=4        <154>
10050 : NEXT I2                           <102>
10060 NEXT I1                             <001>
10070 PRINT                               <010>
10080 PRINT"SOLL ICH WEITERMACHEN ?"      <229>
10090 GET I$:IF I$<>"J"AND I$<>"N"THEN 100
      90                                  <112>
10095 IF I$="J"THEN 1125:REM WEITERMACHEN <233>
10099 END                                 <195>
```

Listing 4. Eine pfiffige Maus mit Durchblick

In diesem Kapitel haben Sie einige wichtige Grundlagen der Künstlichen Intelligenz kennengelernt. Ein entscheidender Faktor, der bei einem Lebewesen auf Intelligenz schließen läßt, ist das Erkennen von Fehlern und das Lernen aus Irrtümern. Ein weiterer entscheidender Faktor ist das Zurücknehmen einer Entscheidung, die sich als Fehler erwiesen hat – der Backtracking-Vorgang.

Als nächsten Kapitel schneiden wir erstmals ein ganz anderes Themengebiet der KI an: Programme, die einen menschlichen Gesprächspartner simulieren können. Zur Übung empfehle ich Ihnen an dieser Stelle jedoch erst einmal das Maus-Programm so zu verbessern, daß der Speicher der Maus als relative Datei angelegt wird. Mit Ihrem bisher erworbenen Wissen über Dateiverwaltung und relative Dateien dürfte dies nicht allzu schwierig sein.

## Eliza

```
HI, I AM ELIZA TELL ME YOUR PROBLEM
) I HATE SCHOOL
SAY, DO YOU HAVE ANY PSYCHOLOGICAL PROBLEMS?
)I CAN'T GET NO SATISFACTION
HOW DO YOU KNOW YOU CAN'T GET NO SATISFACTION
)I JUST DON'T HAVE ANY
WHAT DOES THAT SUGGEST TO YOU?
)I NEED GIRLS AND DRUGS AND ROCKN ROLL
I SEE
)CAN YOU GIVE ME DRUGS?
DON'T YOU BELIEVE THAT I CAN GIVE  DRUGS!
)I LIKE YOU ELIZA
WE WERE DISCUSSING YOU -- NOT ME
)BY THE WAY - MY NAME IS MIKE
NAMES DON'T INTEREST ME
)DON'T YOU LIKE MY NAME?
OH, I LIKE YOUR NAME
)YOU LIE!
YOU'RE NOT REALLY TALKING ABOUT ME, ARE YOU?
)I AM
DID YOU COME TO ME BECAUSE YOU ARE
)YOU MAKE ME SICK
WE WERE DISCUSSING YOU -- NOT ME
)MAYBE I GET SATISFACTION SOON
YOU DON'T SEEM QUITE CERTAIN
)YOU KNOW TO MUCH
OH, I KNOW TO MUCH
)YES
ARE YOU SURE?
)I HATE YOU!
I'M NOT SURE I UNDERSTAND YOU
)SHUT UP!
COME COME ELUDICATE YOUR THOUGHTS
)WHY DON'T YOU GIVE UP ASKING?
DO YOU REALLY BELIEVE I DON'T GIVE UP ASKING?
)I HOPE SO
CAN YOU ELABORATE ON THAT?
)YES
I SEE
)WHY CAN'T I GET ANY ROCKN ROLL?
DO YOU THINK YOU SHOULD BE ABLE TO GET ANY
ROCKN ROLL?
)HAHAHA
THAT IS QUITE INTERESTING
)YOU ARE STUPID!
WHAT MAKES YOU THINK I AM STUPID?
)YOU GIVE ALWAYS STUPID ANSWERS
YOU'RE NOT REALLY TALKING ABOUT ME, ARE YOU?
)YOUR ANSWERS ARE ABSOLUTELY NONSENS
WHY NONSENS?
)I AM BORN TO BE WILD
HOW LONG HAVE YOU BEEN BORN TO BE WILD?
)SINCE I'VE BEEN BORN
SAY, SO YOU HAVE ANY PSYCHOLOGICAL PROBLEMS?
)CAN'T YOU RECOGNIZE THAT?
WE WERE DISCUSSING YOU -- NOT ME
)I AM NOT THE PSYCHIATRIST
DO YOU BELIEVE IT IS NORMAL TO BE NOT THE
PSYCHIATRIST?
)YOU WON!
OH, I WON
)YES
I UNDERSTAND
)AHHHHHHHHHHHHHHHHH AHHHHHH AHHHHHH AHHHHHHHHH
AHHHHH AHHHHHHHHH
```

Eliza ist ein Programm, das seine Funktionsweise am besten selbst erklärt. Deshalb habe ich zunächst auf eine Einleitung verzichtet und stattdessen einen Dialog aufgeführt, den ich erst vor kurzem mit »Eliza« hatte.

Eliza ist jetzt über 25 Jahre alt. Trotzdem ist dieses Programm auch heute noch ein vorzügliches Beispiel für die Programmierung von Sprache verstehenden KI-Programmen. Eliza wurde in den 60er Jahren von dem Computerwissenschaftler Josef Weizenbaum geschrieben. Ziel des Programms war es, einen Psychiater zu simulieren. Der Erfolg von Eliza war überwältigend – zahlreiche menschliche Gesprächspartner dachten tatsächlich, sie hätten es mit einer denkenden, intelligenten Maschine zu tun. Noch viel erstaunlicher ist die Tatsache, daß das Eliza-Programm nur 8 KByte RAM benötigt!

Kein Wunder, denn vor 25 Jahren dachte noch niemand daran, daß es bereits 1983 Homecomputer mit 64 KByte RAM geben würde. Eliza wurde damals noch auf einem Großrechner betrieben. Erst Ende der 70er Jahre fand Eliza auch in Deutschland weite Verbreitung – auf dem PET 2001 von Commodore, einem der ersten erschwinglichen Homecomputer überhaupt. Der RAM-Speicher des PET war 8 KByte groß – groß genug für die Eliza. Auch ich begegnete Eliza 1980 auf einem PET 2001 bei einem Freund. Heute ist der PET längst Nostalgie. Kaufen kann man ihn nicht mehr und wer ihn hat, gibt ihn einfach nicht mehr her. Zuerst wurde er vom ersten Farb-Homecomputer VC 20 in die Ecke gedrängt, um schließlich vom C 64 endgültig überrollt zu werden. Mit ihm geriet auch ein berühmtes Programm in Vergessenheit – die bereits oben erwähnte Eliza. Auch meine alten PET-Programmkassetten verstaubten schnell in ihren Hüllen, als ich meinen C 64 mit atemberaubenden Action-Games hatte. Erst vor einem halben Jahr fand ich Eliza wieder. Eliza ist eines der ganz wenigen Programme, die auf wirklich allen Commodore-Computern mit 8 KByte laufen. Ich speicherte die Eliza also einfach auf Disk, um sie bei Gelegenheit einigen Freunden auf dem C 64 vorzustellen. Das Resultat war verblüffend: Das Eliza-Programm hat auch heute nichts an Originalität eingebüßt. Nach wie vor versetzt es in Erstaunen und spätestens dann in eine gewisse Ehrfurcht, wenn man hört, wie kurz der Programmcode ist.

Josef Weizenbaum, der Vater von Eliza, hat später noch verbesserte Programmversionen erstellt. Eine davon wurde unter dem Namen DOCTOR bekannt.

Wer heute intelligente Adventures oder gar KI-Programme schreiben will, kommt an dem Programm Eliza nicht vorbei. Zur Einführung in die Programmierung von Programmen, die Sprache verstehen, werden wir deshalb nun zusammen einmal das Eliza-Programm analysieren. Bevor Sie sich jedoch meine nun folgende Theorie zu Gemüte führen, sollten Sie erst einmal mit der Eliza spielen. Das Listing zur Eliza finden Sie in Listing 5.

Zum Programmablauf läßt sich nicht viel sagen. Einfach RUN eingeben und Sie haben einen garantiert außergewöhn-

lich geduldigen Gesprächspartner vor sich. Gute Unterhaltung!

Wenn Sie sich nun eine Weile mit dem Eliza-Programm »unterhalten« haben, haben Sie sicherlich festgestellt, daß nicht selten relativ unsinnige Antworten zustandekommen. Daß im Programm nicht viel Intelligenz stecken kann, sieht man schon am Listing: Das Hauptprogramm besteht aus nur zirka 50 Zeilen, der restliche Teil aus DATA-Zeilen. In der Tat, Eliza hat keine Ahnung von dem Gesprächsinhalt der Dialoge, die sie führt. Der Eliza-Effekt, wie wir es einmal nennen wollen, besteht also darin, aus eingegebenen Sätzen und Antworten neue Fragen zu formulieren. Daß Eliza Antwort gibt, ohne nachzudenken und darauf verzichtet, die Antwort auf Sinngehalt zu untersuchen, zeigt sich dann, wenn man anstelle von einfachen Sätzen wirre Zeichenfolgen eingibt.

Ich muß an dieser Stelle zugeben, daß der Beispielsdialog, den ich zur Einleitung dieses Kursabschnitts gezeigt habe, nur deshalb relativ intelligent erscheint, weil ich durch gewählte Antworten (unter Kenntnis, wie Eliza arbeitet), vermeiden konnte, daß Eliza unsinnige Antworten gibt.

Geben Sie nach Programmstart doch bitte einmal den folgenden Satz ein:

»CAN YOU X«

Eliza antwortet mit »DON'T YOU BELIEVE I CAN DO X«.

Nach dem Schlüsselbegriff CAN YOU hat Eliza also einfach den Satzrest X abgeschnitten und an einen ihrer Standardsätze angehängt. Hätte man als ersten Satz

»CAN YOU DANCE« eingegeben, so hätte Eliza mit »DON'T YOU BELIEVE THAT I CAN DANCE« geantwortet. Geben wir jetzt nochmals »CAN YOU X« ein. Eliza antwortet jetzt mit:

»PERHAPS YOU WOULD LIKE TO BE ABLE TO X«. Beim dritten gleichen Versuch antwortet Eliza schließlich mit »YOU WANT ME TO BE ABLE TO X«. Gibt man zum viertenmal »CAN YOU X« ein, so erhält man wieder die erste Antwort. Eliza hat also nur eine begrenzte Anzahl von Antworten und Fragen zu bestimmten Schlüsselwörtern parat.

Eliza ist also eigentlich gar kein KI-Programm, sondern lediglich eine einfache Gesprächs-Simulation. Trotzdem will ich das Programm ausführlich besprechen, denn es läßt sich vieles aus ihm herausholen. Es bietet sich zum Beispiel sehr an, eine eliza-artige Routine in ein Text-Adventure einzubauen. Man könnte die Schlüsselwörter von Eliza speziell auf das Adventure abrechnen. Der Spieler könnte dann einen richtigen Dialog mit dem Computer führen, der in etwa so aussehen könnte:

Spieler: Ich weiß hier nicht mehr weiter.
ADV: Denken Sie halt mal scharf nach!
Spieler: Ich krieg die verdammte Tür nicht auf!
ADV: Sie brauchen ein Hilfsmittel, wenn Sie die Tür öffnen wollen.
Spieler: Womit läßt sich die Tür öffnen?
ADV: Stellen Sie sich nicht so dumm an.
Spieler: Womit läßt sich die Tür öffnen?
ADV: Sie kommen bestimmt nicht weiter, wenn Sie immer wieder auf der gleichen Frage herumhacken.

Und so kann es noch lange weitergehen.

Dies sieht momentan natürlich wieder viel intelligenter aus, als es in Wirklichkeit ist. Auf jeden Fall bietet es sich an, das Eliza-Programm einmal genau unter die Lupe zu nehmen, um das Prinzip zu verstehen, das hinter dem Bilden der Antworten steht: Die wichtigsten Infomationen werden der vorausgegangenen Frage entnommen. Alle Gesprächspartner-Simulationsprogramme bauen irgendwie auf dem Eliza-Prinzip auf. Es ist also unbedingt erforderlich, die Programmierung der Eliza zu verstehen, wenn man eigene Simulationsprogramme dieser Art schreiben will. Wie gesagt, das Eliza-Programm benötigt lediglich 8 KByte RAM. Das heißt, uns stehen noch 30 KByte Basic-RAM zur Verfügung. Nehmen wir alle 64 KByte RAM und bedienen uns auch noch einer relativen Datei auf Disk, so können wir Programme entwickeln, die den »IQ« der Eliza um ein Hundertfaches übersteigen.

## Eliza unter der Lupe

Das Eliza-Listing läßt sich in zwei große Hauptabschnitte unterteilen:

Dem Steuerprogramm (Zeile 5 bis 620) und DATA-Zeilen (Zeile 1050 bis 2250).

Programmieranfänger sehen im Steuerprogramm lediglich ein Wirrwarr von Stringoperationen. Man muß schon ziemlich genau hinschauen, um zu erkennen, was sich in den einzelnen Zeilen des Steuerprogramms abspielt. Lassen Sie uns dies jetzt einmal gemeinsam tun:

**Eliza: Dokumentation zum Listing**

10 bis 30 Unterprogramm zum Positionieren des Zeigers auf die DATA-Zeilen. Um das Programm möglichst klein zu halten (es stammt schließlich von einem PET 2001 mit nur 8 KByte), war es nicht möglich, die gesamten DATAs in ein Stringfeld einzulesen. Deshalb wird dieses Unterprogramm benutzt, um auf einen bestimmten Satz N zu positionieren. Der DATA-Zeiger wird einfach auf das erste DATA gesetzt (mittels RESTORE), dann werden N-Elemente mittels READ überlesen. Es hat auch seinen guten Grund, daß dieses Unterprogramm am Kopf des Listings steht: Je niedriger die Zeile ist, in der ein Unterprogramm beginnt, umso schneller wird es mit GOSUB aktiviert. Da bei Eliza ständig positioniert werden muß, lohnt es sich aus Geschwindigkeitsgründen, das Unterprogramm voranzustellen.

100 bis 140 Hier wird eine Schlüsselbegriff-Tabelle erzeugt. In die Variablen S%, R%, N% (DIM-Felder) werden die Zahlendatas ab Zeile 2530 eingelesen – also alle Zahlendatas, die es überhaupt gibt. Diese Tabelle ist die wichtigste Tabelle im ganzen Programm (und auch die einzige).

Die gesamte Datei läßt sich in 3 Bereiche gliedern:

Bereich 1 mit N1-Sätzen: Enthält alle Schlüsselwörter, die Eliza kennt.
Bereich 2 mit N2-Sätzen: Hilfsverben
Bereich 3 mit N3-Sätzen: Kommentare

160 Beginn des Dialogs
170 »Befehlseingabe« in den String I$
200 bis 220 Alle ' werden aus I$ herausgekürzt.

Beispiel: Aus »I'M HUNGRY« wird »IM HUNGRY«. Dies ist notwendig, um später I AM und IM gleichsetzen zu können. Im Prinzip ist dieses Kürzen nicht unbedingt notwendig, da in der Schlüsselwort-Tabelle I AM und I'M aufgeführt werden.

230 In P$ ist jeweils die letzte Befehlseingabe gespeichert. Es wird hier dafür gesorgt, daß der Anwender nicht zu schnell das Eliza-Schema begreift, indem er mehrmals hintereinander die gleiche Antwort gibt. Dies gilt allerdings nur bei Verlegenheitsfällen (Erklärung folgt).
280 Der DATA-Zeiger wird auf den ersten Satz des ersten Dateibereiches positioniert.
300 bis 340 Der Befehlsstring I$ wird auf Schlüsselwörter hin durchsucht. Es wird zum Beispiel gesucht, ob »CAN YOU« im String vorkommt.
325 Wird ein Schlüsselwort gefunden, so speichert
S die Nummer des Schlüsselwortes (CAN YOU = 1, CAN I = 2 etc.),
T ab welchem Zeichen von I$ das Schlüsselwort steht.
F$: das Schlüsselwort selbst
365 Prüfen, ob ein Schlüsselwort gefunden wurde (S größer 0). Wurde eines gefunden, so speichert
K: die Nummer des Schlüsselwortes (K=S),

L: wo Schlüsselwort in I$ steht (L=T).
Sprung nach 100

370 Es wurde kein Schlüsselwort gefunden. In diesem Fall erhält K den Wert von N1 (=38). Man kann aber auch sagen, daß doch ein Satz gefunden wurde - Satz 38, der bei uns NOKEYFOUND heißt. Das heißt, immer wenn kein Schlüsselwort gefunden wird, positioniert Eliza auf Satz 38.

400 Ein Schlüsselwort wurde gefunden. N=N1

405 Jetzt wird auf den 2. Bereich positioniert.

410 Nehmen wir einmal an, der Befehlssatz lautete »CAN YOU X«. In diesem Fall nimmt C$ den Wert »X« an.

420 bis 540 Änderung der Bezugswörter. Beispiel: Aus »YOU APPEAR IN MY DREAMS« als C$ wird nach der Umwandlung gemäß Bereich 2 »APPEAR IN YOUR DREAMS«.

560 bis 570 Positionieren auf dritten Datenblock.

580 Positionieren aus SATZ R%(K) des 3. Bereiches. Warum R%(K) und nicht einfach K? Ganz einfach - Eliza bedient sich eines Tricks, um bei gleichen Schlüsselwörtern nicht immer identische Antworten geben zu müssen. Betrachten Sie bitte die Dimensionierungsanweisung im Listing. Es gibt drei Feldvariablen: S%(X), R%(X) und N%(X). Nehmen wir einmal an, Eliza erhält viermal hintereinander das gleiche Schlüsselwort,
zum Beispiel »CAN YOU SWIM«
»CAN YOU DANCE« etc.
Immer lautet das erste Schlüsselwort »CAN YOU«. Das erste Mal antwortet Eliza mit dem Satz R%(1) des 3. Bereiches. Wäre das Schlüsselwort zum Beispiel ARE YOU, so würde sie mit R%(9) antworten.

590 Jetzt wird R%(1) um den Wert 1 erhöht. Überschreitet R%(1) den Wert N%(1), so wird R%(1) auf den Wert S%(1) zurückgesetzt. Das Listing zeigt uns, daß es zum Schlüsselwort CAN YOU drei mögliche Antwortsätze gibt, die in den Sätzen 1 bis 3 stehen. In S%(X) steht immer gespeichert, welcher Satz der erstmögliche Antwortsatz zu einem bestimmten Schlüsselwort ist. In N%(X) ist es der letzte. So wird erreicht, daß nicht allzuschnell auffällt, wie dumm Eliza eigentlich ist.

Nun haben wir auch schon das komplette Programm Eliza dokumentiert. Ich weiß, daß es für einen Anfänger schwierig ist, das Programmkonzept der Eliza komplett zu verstehen. Meine Dokumentation kann Ihnen nur helfen; verstehen werden Sie das Programm jedoch erst dann, wenn Sie es eigenständig Schritt für Schritt durcharbeiten.

Im folgenden möchte ich das Eliza-Konzept noch einmal mit ganz einfachen Worten zusammenfassen:

## Der Eliza-Effekt

Der Eliza-Effekt besteht darin, aus einer Antwort eine neue Frage zu formulieren.

Eliza untersucht eingegebene Sätze auf Schlüsselwörter hin. Nehmen wir einmal an, jemand gibt als Antwortsatz »I HATE SCHOOL« ein. Eliza macht sich nun keineswegs die Mühe, den ganzen Satz in einzelne Wörter zu zerlegen - wie ich es zum Beispiel im ersten Adventure-Sonderheft gemacht habe.

Eliza beruft sich vielmehr zunächst auf die Tabelle der Schlüsselwörter. Zunächst nimmt sie das erste Wort der Tabelle (CAN YOU) als Schlüsselwort beziehungsweise Vergleichsstring K$. Nun wird der gesamte Befehlsstring I$ durchsucht.

```
FORI=1 TO LEN(I$)
: IF MID$(IS,I,LEN(K$))=K$ THEN REM GEFUNDEN
NEXTI
```

Das Suchen erfolgt also einfach durch Durchlaufen des gesamten Befehlsstrings und Vergleichen mit dem Schlüsselwort K$. Dieser Suchalgorithmus ist sehr primitiv; es gibt

einen viel besseren, den ich Ihnen später noch kurz vorstellen werde.

Aus dem Listing können wir lesen, daß Eliza bei dem Befehlsstring »I HATE SCHOLL« mit Sicherheit kein Schlüsselwort finden wird. Daher wird Eliza zu »I HATE SCHOOL« auch keine geistreichen Bemerkungen geben können. Sie bringt eine Ausweichfrage – sie weicht also einfach auf ein anderes Thema aus (wie wir es auch tun, wenn wir nicht mehr weiter wissen). Nach einer erfolglosen Schlüsselwortsuche steht der Tabellenzeiger auf Satz 38 (NOKEYFOUND). Der letzte Satz der Schlüsselworttabelle ist somit immer der Hinweiser auf die Ausweichfragen (zum Beispiel SAY DO YOU HAVE ANY PSYCHOLOGICAL PROBLEMS) oder Verlegenheitsantworten (I SEE).

Aus dem Listing können wir erkennen, wo die Ausweichsätze stehen: S%(38) zeigt auf den Beginn und N%(38) auf das Ende beziehungsweise den letzten der Verlegenheitssätze. R%(38) ist, wie bereits dokumentiert, der Zeiger auf den Ausweichbereich. Er wird immer dann um 1 erhöht (auch zurückgestellt wenn größer als N%), wenn Eliza auf den Ausweichbereich zugegriffen hat. Eliza kennt genau 7 Ausweichsätze, die in den Sätzen 106 bis 113 des Bildes 1 stehen. Das heißt, Eliza wiederholt sich bei jeder 7. Ausweichantwort. Von der Größe der einzelnen Bereiche, also dem Antwortenreich-

```
5 GOTO 100                                    <189>
10 RESTORE                                    <060>
20 FOR I=1 TO N:READ S$:NEXT                  <175>
30 RETURN                                     <088>
100 DIM S%(38),R%(38),N%(38)                  <245>
110 N1=38:N2=12:N3=118                        <032>
120 N=N1+N2+N3:GOSUB 10                       <050>
130 FOR X=1 TO N1                             <028>
140 READ S%(X),L:R%(X)=S%(X):N%(X)=S%(X)+L
    -1:NEXT                                   <134>
160 PRINT"HI, I AM ELIZA TELL ME YOUR PROB
    LEM                                       <235>
170 INPUT I$:I$=" "+I$+" "                    <191>
200 FOR L=1 TO LEN(I$)                        <030>
210 IF MID$(I$,L,1)="'"THEN I$=LEFT$(I$,L-
    1)+RIGHT$(I$,LEN(I$)-L):GOTO 210          <118>
220 NEXT                                      <230>
230 IF I$=P$THEN PRINT"PLEASE DON'T REPEAT
     YOURSELF":GOTO 170                       <202>
280 RESTORE:S=0                               <180>
300 FOR K=1 TO N1                             <096>
305 IF S>0 THEN 340                           <050>
310 READ K$                                   <242>
320 FOR A=1 TO LEN(I$)-LEN(K$)                <151>
325 IF MID$(I$,A,LEN(K$))=K$THEN S=K:T=A:F
    $=K$                                      <015>
330 NEXT                                      <086>
340 A=0:NEXT                                  <218>
365 IF S>0 THEN K=S:L=T:GOTO 400              <227>
370 K=N1:GOTO 560                             <243>
400 N=N1                                      <106>
405 GOSUB 10                                  <089>
410 C$=" "+RIGHT$(I$,LEN(I$)-LEN(F$)-L+1)     <142>
420 FOR X=1 TO N2/2:READ S$,R$                <208>
440 FOR L=1 TO LEN(C$)                        <004>
450 IF L+LEN(S$)>LEN(C$)THEN 510              <096>
480 IF MID$(C$,L,LEN(S$))<>S$THEN 510         <002>
490 C$=LEFT$(C$,L-1)+R$+RIGHT$(C$,LEN(C$)-
    L-LEN(S$)+1):L=L+LEN(R$)                  <093>
500 GOTO 540                                  <014>
510 IF L+LEN(R$)>LEN(C$)THEN 540              <040>
520 IF MID$(C$,L,LEN(R$))<>R$THEN 540         <217>
530 C$=LEFT$(C$,L-1)+S$+RIGHT$(C$,LEN(C$)-
    L-LEN(R$)+1)                              <161>
540 NEXT:NEXT                                 <035>
555 IF MID$(C$,2,1)=" "THEN C$=RIGHT$(C $,
    LEN(C$)-1)                                <241>
560 N=N1+N2                                   <015>
570 GOSUB 10                                  <000>
580 FOR X=1 TO R%(K):READ F$:NEXT X           <116>
590 R%(K)=R%(K)+1:IF R%(K)>N%(K)THEN R%(K)
    =S%(K)                                    <114>
600 IF RIGHT$(F$,1)<>"*"THEN PRINT F$:P$=I
    $:GOTO 170                                <000>
620 PRINT LEFT$(F$,LEN(F$)-1);C$:GOTO 170     <126>
1050 DATA"CAN YOU","CAN I","YOU ARE","YOUR
     E","I DONT","I FEEL","WHY DONT YOU"      <101>
1060 DATA"WHY CANT I","ARE YOU","I CANT","
     I AM","IM ","YOU ","I WANT","WHAT"       <036>
1070 DATA"HOW","WHO","WHERE","WHEN","WHY",
     "NAME","CAUSE","SORRY","DRAEM","HELLO
     "                                        <074>
1080 DATA"HI ","MAYBE"," NO","YOUR","ALWAY
     S","THINK","ALIKE","YES","FRIEND"        <034>
1090 DATA"COMPUTER","ELIZA","MONEY","NOKEY
     FOUND"                                   <209>
1230 DATA" ARE "," AM ","WERE ","WAS "," Y
     OU "," I ","YOUR ","MY "," IVE "         <208>
1235 DATA" YOUVE "," IM "," YOURE "           <045>
1330 DATA"DON'T YOU BELIEVE THAT I CAN*"      <156>
1340 DATA"PERHAPS YOU WOULD LIKE TO BE ABL
     E TO*"                                   <249>
1350 DATA"YOU WANT ME TO BE ABLE TO*"         <113>
1360 DATA"PERHAPS YOU DON'T WANT TO*","DO
     YOU WANT TO BE ABLE TO*"                 <167>
1365 DATA"WHAT MAKES YOU THINK I AM*","DOE
     S IT PLEASE YOU TO BELIEVE I AM*"        <224>
1390 DATA"PERHAPS YOU WOULD LIKE TO BE*"      <065>
1400 DATA"DO YOU SOMETIMES WISH YOU WERE*"    <219>
1410 DATA"DON'T YOU REALLY*","WHY DON'T YO
     U*","DO YOU WISH TO BE ABLE TO*"         <059>
1440 DATA"DOES THAT TROUBLE YOU?","TELL ME
      MORE ABOUT SUCH FEELINGS."              <157>
1460 DATA"DO YOU OFTEN FEEL*","DO YOU ENJO
     Y FEELING*"                              <039>
1480 DATA"DO YOU REALLY BELIEVE I DON'T*",
     "PERHAPS IN GOOD TIME I WILL*"           <137>
1500 DATA"DO YOU WANT ME TO*","DO YOU THIN
     K YOU SHOULD BE ABLE TO*"                <002>
1520 DATA"WHY CAN'T YOU*","WHY ARE YOU INT
     ERESTED IN WHETHER OR NOT I AM*"         <159>
1540 DATA"WOULD YOU PREFER IF I WHERE NOT*
     ","PERHAPS IN YOUR FANTASIES I AM*"      <020>
1560 DATA"HOW DO YOU KNOW YOU CAN'T*","HAV
     E YOU TRIED?","PERHAPS YOU CAN NOW*"     <247>
1590 DATA"DID YOU COME TO ME BECAUSE YOU A
     RE*","HOW LONG HAVE YOU BEEN*"           <255>
1610 DATA"DO YOU BELIEVE IT IS NORMAL TO B
     E*","DO YOU ENJOY BEING*"                <229>
1630 DATA"WE WERE DISCUSSING YOU-- NOT ME.
     ","OH, I*"                               <229>
1650 DATA"YOU'RE NOT REALLY TALKING ABOUT
     ME, ARE YOU?"                            <015>
1660 DATA"WHAT WOULD IT MEAN TO YOU IF YOU
      GOT*","WHY DO YOU WANT*"                <116>
1680 DATA"SUPPOSE YOU SOON GOT*","WHAT IF
     YOU NEVER GOT*"                          <099>
1700 DATA"I SOMETIMES ALSO WANT*","WHY DO
     YOU ASK?"                                <189>
1720 DATA"DOES THAT QUESTION INTEREST YOU?
     "                                        <086>
1730 DATA"WHAT ANSWER WOULD PLEASE YOU THE
      MOST?"                                  <076>
1740 DATA"WHAT DO YOU THINK?","ARE SUCH QU
     ESTIONS IN YOUR MIND OFTEN?"             <078>
1760 DATA"WHAT IS THAT YOU REALLY WANT TO
     KNOW?","HAVE YOU ASKED ANYONE ELSE?"     <188>
1780 DATA"HAVE YOU ASKED SUCH QUESTIONS BE
     FORE?"                                   <023>
1790 DATA"WHAT ELSE COMES TO MIND WHEN YOU
      ASK THAT?"                              <128>
1800 DATA"NAMES DON'T INTEREST ME.","I DON
     'T CARE ABOUT NAMES-- PLEASE GO ON."     <240>
1820 DATA"IS THAT THE REAL REASON?","DON'T
      ANY OTHER REASONS COME TO MIND?"        <239>
1840 DATA"DOES THAT REASON EXPLAIN ANYTHIN
     G ELSE"                                  <231>
1850 DATA"WHAT OTHER REASONS MIGHT THERE B
     E?","PLEASE DON'T APOLOGIZE!"            <120>
1870 DATA"APOLOGIES ARE NOT NECESSARY."       <196>
1880 DATA"WHAT FEELINGS DO YOU HAVE WHEN Y
     OU APOLOGIZE?","DON'T BE SO DEFENSIVE
     !"                                       <001>
1900 DATA"WHAT DOES THAT DREAM SUGGEST YOU
     ?","DO YOU DREAM OFTEN?"                 <134>
1920 DATA"WHAT PERSONS APPEAR IN YOUR DREA
     MS?","ARE YOU DISTURBED BY YOUR DREAM
     S?"                                      <171>
1940 DATA"HOW DO YOU DO ... PLAESE STATE Y
```

**Listing 5. Das Eliza-Programm**

tum eines Eliza-Programms, hängt es ab, wie schnell die eigentliche »Dummheit des Programmes« vom Spieler erkannt wird.

Hat Eliza beim ersten Durchlauf also kein Schlüsselwort gefunden, so bricht sie alle weiteren »Berechnungen« ab und antwortet sofort mit einer Verlegenheitsantwort oder stellt eine Ausweichfrage. Dies kann man mit einem Adventure-Parser vergleichen, der abbricht, wenn er ein Wort nicht kennt – ICH KENNE DAS WORT X NICHT.

Lassen Sie uns jetzt jedoch einmal von einem interessanten Befehlssatz ausgehen. Unter Befehlssatz verstehe ich alle Arten von Sätzen, die ein Spieler beziehungsweise Anwender auf einen INPUT-Befehl hin eingibt. So zum Beispiel »YOU APPEAR IN MY DREAMS«. Eliza findet nun das Schlüsselwort »YOU« – Satz 13 in Bereich 2. Anhand des Listings können wir herausfinden, daß Eliza mit »We were discussing you – not me« antworten muß. Sicher haben Sie bereits bemerkt, daß im Listing manche Sätze mit einem Sternchen enden und andere wiederum nicht. Sätze, die nicht mit einem Sternchen enden, werden direkt ausgegeben, also so wie sie im Listing stehen. Der Antwortsatz »WE WERE DISCUSSING YOU – NOT ME« steht dort an Stelle 32 und hat am Ende kein Sternchen. Wie wir bereits wissen, steht der YOU-Zeiger jetzt auf Satz 33. Bei der nächsten Satzeingabe muß also als Antwort Satz 33 kommen – wenn im Befehlssatz wieder das Schlüsselwort YOU gefunden wird. Probieren wir dies doch einfach einmal aus. Nach Eingabe von »YOU APPEAR IN MY NIGHTMARES« folgt in der Tat die Antwort »OH, I APPEAR IN YOUR NIGHTMARES«. Der Satz lautet jedoch nur »OH, I« und ist mit einem Sternchen versehen. Bei der Antwort wurde zu Satz 33 also »APPEAR IN YOUR NIGHTMARES« zugefügt. Vergleichen wir nun einmal Eingabe und Ausgabe:

Eingabe: YOU APPEAR IN MY NIGHTMARES
Ausgabe: OH, I APPEAR IN YOUR NIGHTMARES

Hier sieht man ganz deutlich, was passiert ist. Eliza hat vom Eingabesatz den Satzbereich nach dem Schlüsselwort YOU abgetrennt und anschließend an den Ausgabesatz 33 OH, I angefügt. Dieser abgetrennte Bereich taucht im Programm als C$ auf. Bevor C$ jedoch an Satz 33 angefügt wurde, veränderte sich noch etwas: Gemäß Bereich 2 wurde das Wort MY durch YOUR ersetzt.

Damit ist der Eliza-Effekt auch schon beschrieben.

Mit dem jetzt erworbenen Wissen sollte es Ihnen nicht mehr allzu schwer fallen, das Eliza-Programm selbst unter die Lupe zu nehmen.

Die Eliza-Methoden sind zwar sehr simpel, zeigen bei einer Analyse des Programms jedoch ganz einleuchtend, wie man Probleme der Sprachverarbeitung, zum Beispiel Grundlagen der Stringprogrammierung oder der Zerlegungsverfahren für Zeichenketten, angehen kann.

```
     OUR PROBLEM."                          <037>
1950 DATA"YOU DON'T SEEM QUITE CERTAIN.","
     WHY THE UNCERTAIN TONE?"               <199>
1970 DATA"CAN'T YOU BE MORE POSITIVE?","YO
     U AREN'T SURE?","DON'T YOU KNOW?"      <223>
2000 DATA"WHY NO*","DON'T SAY NO IT'S ALWA
     YS SO NEGATIVE","WHY NOT?"             <073>
2030 DATA"ARE YOU SURE?","WHY NO?","WHY AR
     E YOU CONCERNED ABOUT MY*"             <214>
2060 DATA"WHAT ABOUT YOUR OWN*","CAN'T YOU
      THINK OF A SPECIFIC EXAMPLE?","WHEN?
     "                                      <137>
2090 DATA"WHAT ARE YOU THINKING OF?","REAL
     LY, ALWAYS?"                           <076>
2110 DATA"DO YOU REALLY THINK SO?","BUT YO
     U ARE NOT SURE YOU*"                   <060>
2130 DATA"DO YOU DOUBT YOU*","IN WHAT WAY?
     ","WHAT RESEMBLANCE DO YOU SEE?"       <131>
2160 DATA"WHAT DOES THE SIMILARITY SUGGEST
      TO YOU?"                              <241>
2170 DATA"WHAT OTHER CONNECTIONS DO YOU SE
     E?"                                    <160>
2180 DATA"COULD THERE REALLY BE SOME CONNE
     CTIONS?","HOW?"                        <204>
2200 DATA"YOU SEEM QUITE POSITIVE.","ARE Y
     OU SURE?","I SEE.","I UNDERSTAND."     <074>
2240 DATA"WHY DO YOU BRING UP THE TOPIC OF
      FRIENDS?","DO YOUR FRIENDS WORRY YOU
     ?"                                     <038>
2260 DATA"DO YOUR FRIENDS PICK ON YOU?","A
     RE YOU SURE YOU HAVE ANY FRIENDS?"     <154>
2280 DATA"DO YOU IMPOSE ON YOUR FRIENDS?"  <004>
2290 DATA"PERHAPS YOUR LOVE FOR FRIENDS WO
     RRIES YOU.","DO COMPUTERS WORRY YOU?" <242>
2310 DATA"ARE YOU TALKING ABOUT ME IN PART
     ICULAR?"                               <057>
2320 DATA"ARE YOU FRIGHTENED BY MACHINES?"
     ,"WHY DO YOU MENTION COMPUTERS?"       <170>
2340 DATA"WHAT DO YOU THINK MACHINES HAVE
     TO DO WITH YOUR PROBLEM?"              <063>
2350 DATA"DON'T YOU THINK COMPUTERS CAN HE
     LP PEOPLE?"                            <214>
2360 DATA"WHAT IS IT ABOUT MACHINES THAT W
     ORRIES YOU?"                           <087>
2370 DATA"SAY, DO YOU HAVE ANY PSYCHOLOGIC
     AL PROBLEMS?"                          <080>
2380 DATA"WHAT DOES THAT SUGGEST TO YOU","
     I SEE","I'M NOT SURE I UNDERSTAND YOU
     ."                                     <129>
2410 DATA"COME COME ELUCIDATE YOUR THOUGHT
     S","CAN YOU ELABORATE ON THAT?"        <239>
2430 DATA"THAT IS QUITE INTERESTING.","WHY
      DO YOU HAVE PROBLEMS WITH MONEY?"     <054>
2450 DATA"DO YOU THINK MONEY IS EVERYTHING
     ?"                                     <189>
2460 DATA"ARE YOU SURE THAT MONEY IS THE P
     ROBLEM?"                               <235>
2470 DATA"I THINK WE WANT TO TALK ABOUT YO
     U, NOT{2SPACE}ABOUT ME","WHATS ABOUT
     ME?"                                   <151>
2490 DATA"WHY DO YOU ALWAYS BRING UP MY NA
     ME?"                                   <221>
2530 DATA 1,3,4,2,6,4,6,4,10,4,14,3,17,3,2
     0,2,22,3,25,3,28,4,28,4,32,3,35,5,40,
     9                                      <193>
2540 DATA 40,9,40,9,40,9,40,9,40,9,49,2,51
     ,4,55,4,59,4,63,1,63,1,64,5,69,5,74,2 <159>
2550 DATA 76,4,80,3,83,7,90,3,93,6,99,7,11
     6,3,113,3,106,7                        <164>
```

**Listing 5. Das Eliza-Progamm (Schluß)**

## Das Zufallsprinzip – programmierter Zufall

Künstliche Intelligenz ist das, was der Mensch besser kann. So lautet eine der vielen Definitionen, die im Laufe der letzten Zeit entworfen wurden – Versuche, zu beschreiben, was künstliche Intelligenz eigentlich überhaupt ist. Wie ist die obige Definition zu interpretieren? Am besten läßt sie sich wohl so erklären: Stellen wir uns doch einmal einen Arbeiter in einer Farbrik vor. Er hat eine bestimmte vorgegebene Aufgabe, die er an einer Maschine verrichtet. Die Aufgabe könnte zum Beispiel darin bestehen, Werkzeugteile, die auf Fließband A ankommen, auf Fließband B umzuladen. Der Arbeiter steht also zwischen den Fließbändern A und B und verrichtet seine Arbeit. Viel nachzudenken braucht er nicht, um die Aufgabe richtig zu erfüllen – er kennt den Handgriff genau, der notwendig ist: Das Werkzeugteil von Band A nehmen und auf Band B legen. Dieser Handgriff wiederholt sich wieder und wieder. Der Arbeiter ist also gewissermaßen auf eine Tätigkeit programmiert. Stellen wir uns nun einmal einen Roboter, oder genauer gesagt einen Roboterarm vor, der die gleiche Arbeit verrichtet. Auch er hat bei korrekter Programmierung keine Probleme beim Erfüllen des Auftrags. Der Arbeiter hat dem Roboter gegenüber einen gewaltigen Vorteil. Er ist in der Lage, darüber nachzudenken, was er gerade tut. Vielleicht überlegt er, ob er für seine monotone Arbeit ausreichend Lohn erhält. Wenn ihm die Arbeit schließlich zu dumm wird, wenn er »keinen Bock« mehr hat, so kann er die

Arbeit hinwerfen, wobei er sich sogar klar darüber ist, welche Folgen das für ihn haben kann. Unser Arbeiter kann aus dem System aussteigen – er kann aus seiner aufgetragenen Funktion ausbrechen, aus seiner Programmierung aussteigen. Vom Roboter ist eine solche Reaktion niemals zu erwarten – er arbeitet solange weiter, bis ein mechanischer Defekt auftritt oder bis man ihm eine neue Aufgabe gibt. Wer also Programme schreiben will, die möglichst realistisch menschliches Verhalten oder sogar Intelligenz simulieren sollen, der darf diesen Faktor des aus-dem-System-aussteigen-könnens nicht vernachlässigen. Und hier stellt sich die Frage, wie man nun ein solches Aussteigen aus dem System programmieren kann. Am einfachsten wäre es mit einem Basic-Befehl wie:

IF KEIN BOCK MEHR THEN MACH, WAS DU WILLST.

Einen solchen Befehl gibt es jedoch nicht, und es wird ihn höchstwahrscheinlich auch niemals geben. Es gibt jedoch eine andere Lösung...

Schließlich geht es uns ja nur um eine realistische Simulation menschlichen Verhaltens. Sehen wir doch einmal Bild 9 an.

Jeder hat dieses Black-Box-Schema bestimmt schon einmal in irgendeinem Schullehrbuch gesehen. Die Begriffe »Eingabe« und »Ausgabe« sind sehr allgemein zu verstehen. Eine Eingabe kann zum Beispiel eine Zahl, ein Satz oder gar eine ganze Tabelle sein, die zur Bearbeitung eingegeben wird. Was passiert aber nun mit der Eingabe, und wie kommt es zu einer bestimmten Ausgabe? Hierfür steht die Black-Box, das ist der Kasten mit dem Fragezeichen. Am besten läßt sich diese Frage anhand von Bild 10 beantworten: Dieses Struktogramm ist typisch für ein Programm, das menschliches Verhalten simulieren soll.

Auf den ersten Blick ist gleich festzustellen, daß es zur Bearbeitung einer Eingabe nur zwei Möglichkeiten gibt. Entweder geht das Programm logisch vor, das heißt, das Programm berechnet die Ausgabe, die als Antwort auf eine bestimmte Frage am logischsten zu sein scheint. Hier wird

EINGABE ⇒ ? ⇒ AUSGABE

Bild 9. Das Black-Box-Schema

EINGABE

X = INT(100 x RND(1)) + 1

X > ZU | X ≦ ZU

Logisches Vorgehen
Taktischer Zug
Intelligenz

Antwort wird völlig zufällig, ohne Rücksicht auf die Eingabe ermittelt.

AUSGABE

Bild 10. Wie entscheide ich mich – formal

```
10 REM **********************************      <060>
11 REM *   SCHUELER - SIMULATION        *      <100>
12 REM * <C> 1986 BEI MICHAEL NICKLES *        <170>
13 REM **********************************      <063>
100 PRINT"{CLR,YELLOW,RVSON}SCHUELER-SIMUL
    ATION{RVOFF}"                              <171>
110 INPUT"{2DOWN}ZUFALLSANTEIL";ZU             <233>
120 IF ZU<0 OR ZU>100 THEN 110                 <021>
130 PRINT"{2DOWN}LOGIGANTEIL{2SPACE}:"100-
    ZU"%"                                      <120>
140 PRINT"{DOWN}ZUFALLSANTEIL:"ZU"%"           <116>
150 PRINT"{3DOWN}BITTE EINE TASTE DRUECKEN
    !":POKE 198,0:WAIT 198,1                   <165>
1000 REM AUFGABE STELLEN ------------------
     --------------------------------------
     -                                         <022>
1001 :                                         <215>
1010 A=INT(10*RND(1))+1                        <155>
1020 B=INT(10*RND(1))+1                        <169>
1030 PRINT"WIEVIEL IST"A"*"B"?{3SPACE}";       <018>
2000 REM ANTWORT DES SCHUELERS ------------
     --------------------------------------
     -                                         <212>
2001 :                                         <199>
2010 X=INT(100*RND(1))+1                       <098>
2020 IF X>ZU THEN 2100:REM LOGISCHE ANTWOR
     T                                         <140>
2030 IF X<=ZU THEN 2200:REM ZUFALLSANTWORT     <110>
2100 REM RICHTIGE ANTWORT ----------           <063>
2101 :                                         <045>
2110 C=A*B                                     <041>
2120 PRINT C                                   <216>
2130 POKE 198,0:WAIT 198,1:GOTO 1000           <146>
2200 REM ZUFALLSANTWORT ------------           <120>
2201 :                                         <145>
2210 C=INT(100*RND(1))+1                       <216>
2220 PRINT C                                   <060>
2230 POKE 198,0:WAIT 198,1:GOTO 1000           <246>
```

Listing 6. Ein Schüler, der rechnen kann

zum Beispiel der bestmögliche Zug bei einer Schachstellung berechnet. Oder das Programm geht unlogisch vor. Hier gibt das Programm eine Antwort vor, die unabhängig von der Eingabe berechnet wird.

Nehmen wir an, wir unterhalten uns mit einem Programm, zum Beispiel mit ELIZA, und dabei entsteht folgender Dialog:

Mensch: Ich habe Angst vor großen Hunden.

Computer: Finden Sie nicht auch, daß das Wetter heute besonders schlecht ist?

Unter die zweite Kategorie fallen jedoch nur solche Antworten, die wirklich absolut zufällig entstanden sind. Gibt nämlich ELIZA eine Verlegenheitsantwort, weil sie im Eingabesatz kein Schlüsselwort findet, so zählt dies als logisches Vorgehen! Am interessantesten in Bild 10 ist die Formel, mit der ermittelt wird, ob die Antwort auf eine Eingabe zufällig oder per Berechnung folgerichtig erfolgt.

Hierzu dient eine einfache RND-Funktion, mit der eine Zufallszahl erzeugt wird, die im Bereich zwischen 1 und 100 liegt:

`X=INT(100*RND(1))+1`

In der Variablen ZU wird angegeben, wie groß der Zufallsanteil in der »Black-Box« ist.

Nehmen wir einmal an, der Zufallsanteil beträgt 0 Prozent. Es soll also jede Eingabe logisch beantwortet werden

`INT(100*RND(1))+1`

ist aber immer größer als 0! So wird also immer die linke Spalte angewählt. Hat die Variable ZU hingegen den Wert 100 Prozent, so wird

`INT(100*RND(1))+1`

immer kleiner als 100 – die Antwort wird immer zufälliger ermittelt. Mittels dieser Zufallsformel ist es also möglich, das Logik/Zufalls-Verhältnis genau festzulegen.

Das folgende Schülerprogramm (Listing 6) arbeitet nach diesem Prinzip. Es wird ein Schüler simuliert, der nach dem kleinen Einmaleins abgefragt wird. Der Computer stellt die

**Bild 11. Das Domizil von Pacman**

Aufgaben automatisch – man muß nach jeder Antwort einfach eine Taste drücken. Zu Beginn des Programms läßt sich ein Wert für ZU, nämlich der Intelligenzgrad des Schülers, eingeben. Je kleiner der Intelligenzanteil (also je größer ZU ist), um so öfter gibt der Schüler eine falsche Antwort.

Probieren Sie nach Eingabe des Programmes einmal verschiedene ZU-Werte aus.

Es läßt sich mit diesem Programm jede Art von Schüler simulieren – vom Dummkopf bis zur Intelligenzbestie.

Für das Programmieren von Schüler-Simulationen lassen sich folgende Regeln festlegen:

1. Regel: Kein Mensch ist perfekt. Also darf auch eine Simulation niemals vom perfekten Menschen ausgehen. Es ist also falsch, bei der obigen Schülersimulation dem Zufall 0 Prozent Anteil zuzuordnen. Dann wäre der Schüler nämlich hundertprozentig irrtumsfrei. Lassen Sie deshalb auch zur Simulation einer Intelligenzbestie den Zufallsanteil mindestens 1 betragen!

2. Regel: Gehen wir nun noch mal davon aus, daß ein extrem intelligenter Schüler simuliert werden soll. Die erste Regel besagt ja, daß ZU hier mindestens 1 betragen soll. Dies hat zur Folge, daß der Schüler in 99 Prozent der Fälle eine richtige Antwort gibt. Tritt jedoch das eine Prozent ZU in Kraft, so kann es zu einer absolut idiotischen Antwort kommen, wie zum Beispiel 1 mal 1 = 78. Sehr gute Schüler können sich schon mal irren, jedoch nicht derartig extrem. Auch eine falsche Antwort soll bei einer Extrem-guter-Schüler-Simulation deshalb niemals den Eindruck von Dummheit, sondern vielmehr den Eindruck eines Irrtums, also einer versehentlich falschen Antwort erwecken. Man muß also für diesen 1-Prozent-Fall im Programm eine falsche Antwort vorsehen, die nur geringfügig vom richtigen Ergebnis abweicht. Man kann zum Beispiel die Zeile 2210 so abändern:

```
2210 C=A*B: C=C+ INT(6*RND(1))-3
```

Jetzt liegen auch falsche Antworten in der Nähe der richtigen.

3. Regel: ZU darf nicht konstant sein. Wird ein Schüler sehr lange abgefragt, so treten irgendwann Ermüdungserscheinungen auf – die Leistungsfähigkeit geht zurück, ZU nimmt zu.

Die drei Regeln sind äußerst simpel. Sie können bei bestimmten Grenzfällen auch kombiniert werden.

Noch viel anschaulicher wird die Bedeutung des programmierten Zufalls beim Pacman-Algorithmus:

Jeder kennt das Labyrinth in Bild 11, das Zuhaus von Pacman und seinen Gegnern, den vier Gespenstern. Pacman war das erste Videospiel, das von Pong und Space Invader Schema abwich. Die meisten von uns haben sie miterlebt, die Pacman-Welle, die zirka 1979 über die ganze Video-Spielwelt schwappte. In der Tat, Pacman war das erste Videospiel, dessen Programmierung »intelligente« Rechenalgorithmen erforderte. Im Gegensatz zur primitiven Links-rechts-Bewegung der Space Invader war es hier erforderlich, vier Gespenster

(mehr oder weniger intelligent) durch ein Labyrinth laufen zu lassen; und das mit dem Ziel, den Pacman des Spielers zu fangen. Von kaum einem anderen Videospiel gab es soviele Programmversionen wie von Pacman: Pac man, Munchy, Puckman, MS. Pacman und Side Pacman, um nur einige von ihnen zu nennen. Viele haben bestimmt auch schon einmal versucht, eine eigene Pacman-Version (sei es auch nur in Basic) zu programmieren. Das ist gar nicht so leicht! Sind die

```
0 REM *********************************         <050>
1 REM *      PAC-MAN KI-PROGRAMM      *         <050>
2 REM * <C> 1985 BEI MICHAEL NICKLES *          <156>
3 REM *********************************         <053>
4 POKE 53280,0:POKE 53281,0:PRINT"{CLR}";       <098>
10 PRINT"{CLR,YELLOW,RVSON}TAKTIK / ZUFALL
   EXPERIMENT{RVOFF}"                           <092>
20 PRINT"{DOWN}WIE GROSS SOLL DER ZUFALLSA
   NTEIL SEIN                                   <162>
30 INPUT ZU                                     <091>
40 IF ZU<0 OR ZU>100 THEN 10                    <100>
50 PRINT"{2DOWN}ZUFALL:"ZU"%"                   <146>
60 PRINT"{2DOWN}TAKTIK:"100-ZU"%"               <064>
70 PRINT"{3DOWN}BITTE EINE TASTE DRUECKEN!
   "                                            <004>
80 POKE 198,0:WAIT 198,1                        <052>
99 GOSUB 10000                                  <079>
100 X=17:Y=16:XX=1024+X+Y*40:XP=PEEK(XX):P
    OKE XX,42                                   <127>
110 GG=1361:GP=PEEK(GG):POKE GG,160:GW=-1       <178>
1000 REM STEUERUNG SPIELFIGUR -------------
     --------------------------------------
     -                                          <235>
1001 :                                          <215>
1010 JO=56320                                   <036>
1020 IF PEEK(JO)=126 THEN WY=-1:WX=0            <174>
1030 IF PEEK(JO)=125 THEN WY=1 :WX=0            <017>
1040 IF PEEK(JO)=123 THEN WY=0 :WX=-1           <197>
1050 IF PEEK(JO)=119 THEN WY=0 :WX=1            <096>
1100 POKE XX,XP                                 <188>
1110 X=X+WX:Y=Y+WY                              <153>
1120 XX=1024+X+40*Y                             <096>
1130 XP=PEEK(XX)                                <184>
1140 IF XP=102 THEN X=X-WX:Y=Y-WY:GOTO 112
     0                                          <119>
1150 IF X=-1 AND Y=10 THEN X=33:Y=10:GOTO
     1120                                       <238>
1160 IF X=34 AND Y=10 THEN X=0:Y=10:GOTO 1
     120                                        <025>
1200 XP=PEEK(XX)                                <254>
1210 POKE XX,42                                 <018>
2000 REM STEUERUNG GESPENSTER -------------
     --------------------------------------
     -                                          <088>
2001 :                                          <199>
2032 IF GW<>40 THEN IF PEEK(GG-40)=32 OR P
     EEK(GG-40)=96 THEN 2100                    <146>
2034 IF GW<>-40 THEN IF PEEK(GG+40)=32 OR
     PEEK(GG+40)=96 THEN 2100                   <054>
2035 IF GW<>1 THEN IF PEEK(GG-1)=32 OR PEE
     K(GG-1)=96 THEN 2100                       <109>
2037 IF GW<>-1 THEN IF PEEK(GG+1)=32 OR PE
     EK(GG+1)=96 THEN 2100                      <122>
2040 POKE GG,GP:GG=GG+GW                        <010>
2042 IF GG=1459 THEN GG=1424:GW=1 :REM TUN
     NEL                                        <220>
2043 IF GG=1423 THEN GG=1458:GW=-1:REM TUN
     NEL                                        <205>
2045 IF GG=XX THEN POKE 53280,1:END             <181>
2050 IF PEEK(GG)<>32 THEN GG=GG-GW:GOTO 21
     00                                         <105>
2070 GP=PEEK(GG):POKE GG,160                    <199>
2080 GOTO 1000                                  <105>
2100 REM NEUE RICHTUNG WAEHLEN ------------
     --------------------------------------     <054>
2101 :POKE GG,GP                                <067>
2110 REM WELCHE RICHTUNGEN SIND MOEGLICH        <231>
2120 RZ=0:FOR I=0 TO 3:RI(I)=0:NEXT I           <184>
2130 IF GW<>40 THEN IF PEEK(GG-40)=96 OR P
     EEK(GG-40)=32 THEN RI(RZ)=-40:RZ=RZ+1      <013>
2140 IF GW<>-40 THEN IF PEEK(GG+40)=96 OR
     PEEK(GG+40)=32 THEN RI(RZ)=+40:RZ=RZ+
     1                                          <209>
2150 IF GW<>1 THEN IF PEEK(GG-1)=96 OR PEE
     K(GG-1)=32 THEN RI(RZ)=-1 :RZ=RZ+1         <053>
2160 IF GW<>-1 THEN IF PEEK(GG+1)=96 OR PE
     EK(GG+1)=32 THEN RI(RZ)=+1 :RZ=RZ+1        <162>
2200 REM WAHL DES BESTEN ZUGES ------------
     --------------------------------------
     -                                          <084>
2201 :                                          <145>
2205 REM  FORI=1TO4:PRINT" ";RI(I);:NEXT:P
     OKEGG,87:PRINTPEEK(GG+40):WAIT198,1        <222>
2210 GY=INT((GG-1024)/40)                       <172>
2220 GX=GG-1024-GY*40                           <048>
2230 FOR I=0 TO RZ                              <243>
2235 : IF GX=X THEN 2255                        <053>
2240 : IF RI(I)=1 AND GX<X    THEN GW=1         <176>
2250 : IF RI(I)=-1 AND GX>X   THEN GW=-1        <025>
2255 : IF GY=Y THEN 2280                        <177>
2260 : IF RI(I)=40 AND GY<Y   THEN GW=40        <080>
2270 : IF RI(I)=-40 AND GY>Y  THEN GW=-40       <186>
2280 NEXT I                                     <076>
2300 REM ZUFALLS-ELEMENT ------------------
     --------------------------------------
     -                                          <096>
2301 :                                          <245>
2310 IF INT(100*RND(1))+1>ZU THEN 2400          <200>
2320 REM ZUFALLSZUG                             <062>
2330 GW=RI(INT(RZ)*RND(1))                      <078>
2400 REM AUSFUEHREN DES ZUGES -------------
     --------------------------------------
     -                                          <186>
2401 :                                          <091>
2404 GG=GG+GW                                   <251>
2405 IF GG=1459 THEN GG=1424:GW=1 :REM TUN
     NEL                                        <075>
2406 IF GG=1423 THEN GG=1458:GW=-1:REM TUN
     NEL                                        <060>
2410 IF GG=XX THEN POKE 53280,1:END             <038>
2420 GP=PEEK(GG):POKE GG,160                    <039>
2430 GOTO 1000                                  <084>
10000 REM UP-SPIELFELD ZEICHNEN ----------
      ------------------------------------
      --                                        <106>
10001 :                                         <071>
10010 PRINT"{CLR,LIG.BLUE}";                    <144>
10020 DATA"∓∓∓∓∓∓∓∓∓∓∓∓∓∓∓∓∓∓∓∓∓∓∓∓∓∓∓∓∓∓∓
      ∓∓∓∓"                                     <121>
10025 DATA"∓{16SPACE}∓{16SPACE}∓"               <142>
10030 DATA"∓ ∓∓∓∓∓ ∓∓∓∓∓∓∓∓ ∓ ∓∓∓∓∓∓∓∓ ∓∓∓
      ∓∓ ∓"                                     <179>
10035 DATA"∓ ∓∓∓∓∓ ∓∓∓∓∓∓∓∓ ∓ ∓∓∓∓∓∓∓∓ ∓∓∓
      ∓∓ ∓"                                     <184>
10040 DATA"∓{33SPACE}∓"                         <107>
10045 DATA"∓ ∓∓∓∓∓ ∓∓∓ ∓∓∓∓∓∓∓∓∓∓∓ ∓∓∓ ∓∓∓
      ∓∓ ∓"                                     <194>
10050 DATA"∓{7SPACE}∓ ∓{6SPACE}∓{6SPACE}∓
      ∓{7SPACE}∓"                               <135>
10055 DATA"∓∓∓∓∓∓∓ ∓ ∓∓∓∓∓∓ ∓ ∓∓∓∓∓∓ ∓ ∓∓∓
      ∓∓∓∓"                                     <204>
10060 DATA"{6SPACE}∓ ∓ ∓{13SPACE}∓ ∓ ∓{6SP
      ACE}"                                     <110>
10065 DATA"∓∓∓∓∓∓∓ ∓∓∓{SHIFT-SPACE}∓∓∓∓∓∓∓
      ∓∓∓∓{SHIFT-SPACE}∓∓∓ ∓∓∓∓∓∓∓"             <195>
10070 DATA"{12SPACE}∓{YELLOW}SUPERMIKE{LIG
      .BLUE}∓{12SPACE}"                         <144>
10075 DATA"∓∓∓∓∓∓∓ ∓∓∓{SHIFT-SPACE}∓∓∓∓∓∓∓
      ∓∓∓∓{SHIFT-SPACE}∓∓∓ ∓∓∓∓∓∓∓"             <205>
10080 DATA"{6SPACE}∓ ∓ ∓{13SHIFT-SPACE}∓ ∓
       ∓{6SPACE}"                               <098>
10085 DATA"∓∓∓∓∓∓∓ ∓∓∓{SHIFT-SPACE}∓∓∓∓∓∓∓
      ∓∓∓∓{SHIFT-SPACE}∓∓∓ ∓∓∓∓∓∓∓"             <215>
10090 DATA"∓{16SPACE}∓{16SPACE}∓"               <207>
10095 DATA"∓ ∓∓∓∓∓ ∓∓∓∓∓∓∓∓ ∓ ∓∓∓∓∓∓∓∓ ∓∓∓
      ∓∓ ∓"                                     <244>
10100 DATA"∓{5SPACE}∓{21SPACE}∓{5SPACE}∓"       <061>
10105 DATA"∓∓∓∓∓ ∓ ∓∓∓ ∓∓∓∓∓∓∓∓∓∓∓ ∓∓∓ ∓ ∓
      ∓∓∓∓"                                     <254>
10110 DATA"∓{7SPACE}∓ ∓{6SPACE}∓{6SPACE}∓
      ∓{7SPACE}∓"                               <195>
10115 DATA"∓ ∓∓∓∓∓∓∓∓∓∓∓∓∓∓ ∓ ∓∓∓∓∓∓∓∓∓∓∓∓
      ∓∓ ∓"                                     <188>
10120 DATA"∓{33SPACE}∓"                         <187>
10125 DATA"∓∓∓∓∓∓∓∓∓∓∓∓∓∓∓∓∓∓∓∓∓∓∓∓∓∓∓∓∓∓∓
      ∓∓∓∓"                                     <226>
10130 RESTORE                                   <018>
10140 FOR I=1 TO 22:READ I$:PRINT I$:NEXT       <016>
10150 RETURN                                    <046>
```

**Listing 7. Ein Gespenst greift an**

Gespenster zu intelligent, so hat der Spieler kaum eine Chance. Sind sie aber dumm, dann gewinnt der Spieler problemlos, und das Spiel verliert seinen Reiz. Intelligenz (Logik, Berechnung) und Dummheit (Zufallsbewegung) der Gespenster müssen richtig miteinander abgestimmt sein, um einen Spielwitz zu schaffen.

Das Logik-Zufalls-Verhältnis bleibt beim Pacman-Algorithmus übrigens auch bei steigendem Level konstant. Gelangt der Spieler in die nächste Spielrunde, so wird einfach die gesamte Ablaufgeschwindigkeit des Programmes gesteigert. Der Spieler muß dann schneller reagieren und hat weniger Zeit zum Nachdenken – er empfindet die neue Spielrunde schwerer als die zuvor bewältigte. Wie sieht der Algorithmus für ein solches Spiel aus? Lassen Sie uns zunächst davon ausgehen, daß das Gespenst absolut logisch vorgeht. Und

Bild 12. Wenn ein Gespenst zu schlau ist ...

das geht so: Das Gespenst bewegt sich zu Beginn des Spiels in einem beliebigen Gang des Labyrinths in eine beliebige Richtung. Es bewegt sich solange in diese Richtung, bis es an eine Abzweigung gelangt. Dann »überlegt« es sich, welche mögliche Richtung am ehesten zum Spieler führt. In dem Fall, der in Bild 12 vorliegt, würde sich das Gespenst für die Richtung SÜDEN (nach unten) entscheiden.

Verhalten sich alle Gespenster gemäß diesem Schema, so hätte der Spieler, wie bereits gesagt, keine Chance. Wir müssen also den Zufall ins Spiel bringen. Dieser Zufall sorgt dann dafür, daß sich ein Gespenst bei einer Wegrichtung nicht immer logisch verhält und den Weg in Richtung Spieler nimmt, sondern daß es ab und zu auch einen willkürlich gewählten Weg nimmt und diesen solange geht, bis es wieder an eine Kreuzung gelangt. Das folgende Programm in Listing 7 simuliert ein Pacman-Spiel mit einem Gespenst. Der Pacman wir mit einem Joystick in Port 2 gesteuert, die Variable ZU wird zu Beginn des Spiels eingegeben. In dem Spiel Pacman gibt es normalerweise vier Gespenster: Inky, Blinky, Winky und Dinky. In unserem Beispiel programmieren wir erst mal nur ein Gespenst, weil das Basic-Programm sonst viel zu langsam wird.

Probieren Sie bitte auch hier wieder verschiedene ZU-Werte aus. Sie werden schnell feststellen, daß es gar nicht einfach ist, den optimalen ZU-Wert zu finden. Trotzdem kann man auf den Zufallsfaktor in KI-Simulationen und Spielen nicht verzichten, wie es die beiden Beispiele gezeigt haben. Soweit ist ja wohl alles klar. Aber eine Anmerkung möchte ich zum Schluß noch machen: Bei ELIZA gibt es keinen Zufallsfaktor. ELIZA ist jedoch auch keine extrem gute Simulation. Ein Zufallsfaktor ließe sich in das Programm aber leicht einbauen und würde es mit Sicherheit noch verbessern.

(M. Nickles/bs/cg)

# Mein Computer versteht mich

**Die folgenden Seiten beschäftigen sich mit der Programmierung von Parsern – den Programmen, die den Spieler »verstehen«.**

Was ist ein Parser? Schlägt man in einem Englisch-Deutsch-Wörterbuch nach, so findet man unter dem Begriff »to parse«: grammatisch zerlegen, analysieren. Oft wird statt Parser auch der Begriff Phraser: etwas ausdrücken oder formulieren, verwendet. In jedem Parser muß zwangsläufig auch ein Phraser enthalten sein. Der Parser ist in der Lage, eingegebene Sätze (zum Beispiel Adventure-Spiel-Befehle) grammatisch zu zerlegen, sie zu analysieren, um letztendlich eine mehr oder weniger intelligente Antwort zu geben. Es wird also erst »geparst« (Satz zerlegt), dann analysiert (Reaktion auf den Befehl) und schließlich »gephrast« (eine Antwort formuliert). Parser gewinnen im Bereich der Adventures und KI-Programmierung (KI=Künstliche Intelligenz) zunehmend an Bedeutung, denn letztendlich sind sie es schließlich, die eine Kommunikation zwischen Mensch und Maschine mittels der menschlichen Sprache ermöglichen. Wenn man sich einmal vor Augen hält, wie kompliziert die menschliche Ausdrucksweise mittels Sprache sein kann, dann kann man sich auch gut vorstellen, wie schwierig es ist, einen »perfekten« Parser zu programmieren. Trotzdem, die Fortschritte, die in letzter Zeit im Bereich Parser-Programmiertechnik gemacht wurden, sind enorm; man bringt immer bessere Parser in immer weniger Speicherplatz unter. Inzwischen scheinen die Parser-Programmierer jedoch vor ernsten Schwierigkeiten zu stehen, besonders die, die Parser für die deutsche Sprache entwickeln wollen.

## Zweideutigkeiten

Betrachten Sie einmal folgenden Satz:
»Moritz fotografiert die Frau mit dem Fotoapparat.«

Für uns ist der Fall eindeutig: Moritz hat einen Fotoapparat und fotografiert damit eine Frau. Der Satz kann jedoch auch völlig anders gedeutet und trotzdem völlig richtig werden: Moritz fotografiert eine Frau, die einen Fotoapparat bei sich hat. Dies ist nur eines von zahlreichen Beispielen. Uns soll dieses Problem jedoch nicht weiter berühren.

Unser Ziel ist es, einen Parser zu programmieren, der möglichst schnell ist, einen möglichst großen Wortschatz verwalten kann und dem es auch nicht an Intelligenz mangelt.

Der bekannteste Parser, den es gibt, ist der von Infocom. Zunächst möchte ich jedoch noch den Scott-Adams-Parser beschrieben. Scott Adams, Autor von bekannten Adventures wie Adventure, Pirate Cove, Voodoo Castle, The Hulk etc., hält auch heute noch an seinem 2-Wort-Parser fest. Ihm geht es wohl mehr um knifflige Aufgaben und weniger um großartigen Textumfang und Wortschatz. Bei dem 2-Wort-Parser handelt es sich um einen Parser, der nur Sätze versteht, die aus einem Verb und einem Objekt oder einer Richtungsangabe bestehen, zum Beispiel NIMM SCHWERT, WIRF SEIL etc.

Einen 2-Wort-Parser zu programmieren, bereitet wohl kaum jemand allzu große Schwierigkeiten. Auch die meisten Adventures, wie man sie vom Abtippen aus Zeitschriften kennt (zum Beispiel Zauberschloß), bedienen sich eines solchen Parsers. Der nächste Parser, der von sich Reden machte, ist der, der in Melbourne Houses »The Hobbit« zur Geltung kam. Er versteht wirklich komplexe englische Sätze, Verknüpfungen sowie Unterhaltungen mit Personen, die im Spiel auftauchen. Beispiel: »Say to Gandalf please carry me«. Da sich das gesamte Hobbit-Programm auf einmal samt Grafik im Speicher befindet und nie nachlädt, läßt sich auf eine äußerst geschickte Programmierung schließen. Nun aber zum Infocom-Parser. Die Infocom-Adventures waren die ersten, die ständig auf Diskette zugriffen. Dadurch konnte natürlich auch der Parser entsprechend besser gestaltet werden. Infocom-Spiele haben einen Wortschatz von bis zu 1000 Wörtern. Auch die Intelligenz des Parsers läßt nur äußerst selten zu wünschen übrig. Bis vor kurzem galt der Infocom-Parser als das Beste, was sich mit einem 64-KByte-RAM-Computer verwirklichen läßt. Vor noch nicht allzu langer Zeit wurde von Synapse/Broderbond ein Text-Adventure namens Mindwheel veröffentlicht; ausgerüstet mit einem BTZ-(Better than Zork) Parser, der angeblich bis zu 1500 Wörter verstehen soll. An dieser Stelle erlaube ich mir auch gleich zu erwähnen, daß wir im Rahmen dieses Kurses einen BABAZ-Parser (besser als besser als Zork) entwickeln werden. Ich bin mir sicher, daß dieser Parser wirklich das Optimale ist, was sich mit 64 KByte RAM programmieren läßt. Allerdings hängt die Qualität eines Parsers keineswegs von der Größe des Wortschatzes ab. Werfen wir an dieser Stelle doch einmal einen Blick auf mein erstes Adventure »Gordon Saga«. Der Parser – so denke ich heute – war extrem schlecht programmiert. Trotzdem verstand er Sätze wie zum Beispiel »Ich will nach Norden gehen und das Schwert nehmen« genausogut wie »Nimm schnell das Schwert und renne dann sofort nach Norden«. Auch werden Sätze wie »Schließe die Tür« und sogar »Mache schnell die Tür zu« verstanden.

Besonders erstaunlich ist dies alles, wenn man weiß, daß der Wortschatz von Gordon Saga aus lächerlichen 85 Worten besteht. Sie sehen also: Auch mit geringem Wortschatzumfang lassen sich erstaunliche Parser programmieren.

Falls Sie noch keinerlei Vorstellung davon haben, wie ein Parser funktioniert, also wie Befehlssätze zerlegt, codiert und analysiert werden: alles darüber finden Sie im ersten Adventure-Kurs im 64'er-Sonderheft 2/85.

### Der Grundaufbau eines Parsers

Das Grundschema eines jeden Parsers ist in Bild 1 veranschaulicht.

Die Grundlagen der Parsertechnik sind keineswegs spezifisch für den C 64 oder einen 64-KByte-RAM-Computer ausgelegt. Sie gelten für jedes Computersystem – egal, wie groß und wie schnell.

Wie Sie aus Bild 1 entnehmen können, setzt sich der Grundaufbau eines Programmes, das Sprache verstehen soll, aus 4 oder 5 Teilen zusammen. Diese Teile wollen wir nun im einzelnen besprechen:

### Befehlseingabe

Wie der Name schon sagt, geht es hier um die Eingabe eines Befehls. Dieser Befehl muß jedoch nicht unbedingt das Format »TUE DIES UND JENES« haben. Er kann genausogut eine Frage, eine Antwort oder eine Bemerkung sein. Unter einem Befehl versteht man also alles, was einem Computer mitgeteilt wird. Die Befehlseingabe besteht in der Regel beim C 64 aus einer modifizierten INPUT-Routine. Bei anderen Computern, die bei ihrer INPUT-Routine von vorneherein Kommas und Anführungszeichen erlauben, muß erst gar nicht lange modifiziert werden. Eine modifizierte INPUT-Routine für den C 64 stellt Listing 22 (im Artikel Dateiverwaltung) dar. In diesem Bereich des Parser-Programms läßt sich kaum noch etwas verbessern. Eine Befehlseingabe muß jedoch nicht unbedingt über Tastatur erfolgen. Manche Adventures können, zumindest teilweise, mit Joystick oder Maus gesteuert werden. Theoretisch wäre auch eine

Spracheingabe denkbar. Beide Systeme sind aber derart speicherintensiv, daß wir uns nicht näher damit beschäftigen werden.

**Parser**

Von der Befehlseingaberoutine aus gelangt der Befehlssatz (den ich auch in meinem zweiten Programmierkurs stets im String BE$ ablegen will) in den Parser. Der Parser selbst besteht im Prinzip aus zwei Hälften: Befehlszerlegung und Befehlscodierung (siehe Bild 2):

Wer meinen ersten Adventure-Kurs gelesen hat, weiß bereits, was hiermit gemeint ist. In der ersten Hälfte wird der Befehlssatz zunächst in einzelne Worte zerlegt. Nun greift der Parser in der Codierungshälfte auf den Wortschatz zurück und codiert jedes einzelne Wort. Er stellt fest, daß es in seinem Wortschatz nicht enthalten ist. Diese Zerlegung/Codierung läßt sich auf verschiedene Art und Weise durchführen. Hierzu später mehr, denn zunächst müssen wir erst einmal wissen, wie ein Wortschatz aufgebaut ist.

**Wortschatz**

Unter einem Wortschatz versteht man alle Wörter, die ein Adventure beziehungsweise KI-Programm kennt. Bei Eliza jedoch ist der Wortschatz gleich Null – wenn man einmal von den Schlüsselbegriffen und Wörtern absieht. Ich muß an dieser Stelle jedoch auch zugeben, daß sich der Parser meiner Gordon Saga eines kleinen Tricks bedient – er gaukelt dem Spieler mehr Intelligenz vor, als er eigentlich besitzt. Gibt man bei einem Infocom-Spiel beispielsweise den Befehlssatz »Küsse das goldene Schwert« ein, so erhält man mit Sicherheit eine Antwort wie zum Beispiel »Das kann doch nicht Ihr Ernst sein!«. Der Infocom-Parser weiß also, daß es keinen Sinn ergibt, wenn man ein Schwert küßt (Es sei denn, das Schwert ist in Wirklichkeit gar kein Schwert, sondern eine hübsche, verzauberte Prinzessin – lechz – aber lassen wir das lieber!). Woher ein Parser so etwas weiß, werde ich Ihnen später ausführlich erklären!

Betrachten wir noch einmal den Gordon-Saga-Parser: Wenn im Spiel eine Tür vorkommt, so gibt es verschiedene Befehlssatz-Möglichkeiten, um diese Tür zu schließen:

1. SCHLIESSE DIE TÜR.
2. MACHE DIE TÜR ZU.
3. ICH WILL DIE TÜR ZU MACHEN.

Bis heute gibt es keinen anderen deutschen Parser, bei dem eine solche Eingabevielfalt erlaubt ist. Wie schafft der Gordon-Saga-Parser dies, wenn er doch nur über einen Miniwortschatz von 85 Worten verfügt?

Betrachten wir doch einmal einen kleinen Ausschnitt aus dem Wortschatz.

Dort steht gespeichert, daß TÜR ein Objekt ist, dem der Objektwert 1 zugeordnet ist.

Es ist auch festgelegt, daß SCHLIESSE und ZU Verben sind. Da beide die gleiche Bedeutung haben, wird beiden zum Beispiel der Verbwert 8 zugeordnet.

Wenn jetzt einer der 3 obigen Sätze nach den Verben ZU oder SCHLIESSE durchsucht wird, wird immer der Verbenwert (beziehungsweise Verbzahl VE) 8 gefunden.

Bei der Objektsuche wird jedesmal der für TÜR entsprechende Wert 1 gefunden.

Der Parser hat somit zwei Werte gefunden:
Objektzahl = 1
und
Verbzahl = 8

Diese beiden Werte werden dem Analyse-Teil des Parsers übermittelt. Der weiß, daß eine Tür geschlossen werden soll, wenn VE=8 und OB=1. Er überprüft jetzt nur noch zwei Dinge:

- ob überhaupt eine Tür vorhanden ist

und

- ob die Tür nicht bereits zu ist (man kann keine verschlossene Tür schließen).

Schließlich antwortet der Parser mit »Ok. Die Tür ist jetzt zu.«.

Gibt man bei Gordon Saga jedoch zum Beispiel einen Befehl ein, der unsinnig ist und »Schließe das Schwert« lautet, so schweigt der Parser beziehungsweise antwortet mit »Ich verstehe nicht, was Sie wollen«. Er erkennt also nicht – so wie der Infocom-Parser – daß es unsinnig ist, ein Schwert zu schließen. Entschuldigen Sie bitte, wenn ich hier etwas vom eigentlichen Thema Wortschatz abgewichen bin und schon etwas vorausgegriffen habe, aber Sie sehen ja selbst, wie eng die einzelnen Bereiche des Parsers miteinander verschlungen sind! Unser momentanes Problem ist es, wie man einen Wortschatz anlegt und wie man ihn im Computer unterbringt und verwaltet.

Es gibt verschiedene Möglichkeiten, einen Wortschatz zu programmieren. Die Zahl dieser Möglichkeiten hält sich jedoch in überschaubaren Grenzen.

Ganz anders ist dies bei der Analyse. Hier gibt es unheimlich viele Programmiermöglichkeiten. Der Analyse-Teil des Parsers ist nicht zuletzt auch der Teil, in dem die KI steckt. Die KI entscheidet auch, ob der Parser weiß, wovon er redet, oder ob er nur so tut, als verstünde er, was los ist. Sie sehen also schon, daß der »Parser« des ELIZA-Effektes keinesfalls mit einem Parser verglichen werden kann, den ein gutes Adventure haben soll. Wenngleich dies jedoch eine Kombination beider Möglichkeiten nicht ausschließen darf. Im Rahmen dieses Kurses werden alle Parserbereiche ausführlich behandelt. Es werden alle Möglichkeiten beschrieben, die sich mit einem C 64 und Floppy realisieren lassen. Was den Analyse-Teil anbelangt, so kann ich Ihnen jedoch nur einige Lösungsideen vorstellen (Kursteil Adventureprogrammierung). Jeder muß für sich selbst entscheiden, welches Analysekonzept für ihn das beste ist.

**Ausgabe**

Im Ausgabe-Teil des Parsers wird schließlich das Resultat der Analyse ausgegeben. Dies geschieht in der Regel in Form einer PRINT-Ausgabe auf dem Bildschirm.

BEFEHLSZERLEGUNG
BEFEHLSCODIERUNG

Bild 2. Grundfunktionen eines Parsers

BEFEHLSEINGABE
WORTSCHATZ
PARSER
ANALYSE
AUSGABE

Bild 1. Aufbau eines Parsers

Parser 4.0 ist ein völlig Disketten-orientiertes System. Das heißt, der gesamte Wortschatz befindet sich in Form einer relativen Datei auf Diskette. Eine Diskettenseite faßt 170 KByte. Bei dem Wortschatz können wir davon ausgehen, daß die Wörter (Verben, Objekte etc.) eine Wortlänge von 20 Zeichen nicht überschreiten. Man kann theoretisch also Wortschätze mit über 6000 Wörtern verwalten. Praktisch ist dies jedoch Unsinn, da auf der Diskettenseite ja auch noch der Text für das Adventure untergebracht werden muß. Realistisch ist es allerdings, mit Parser 4.0 Wortschätze mit etwa 500 Wörtern zu verwalten – denn dann ist noch genug Platz für Texte. Lassen Sie uns also jetzt gemeinsam den Disk-orientierten Wortparser 4.0 entwickeln. Bevor man allerdings einen solchen Parser programmieren kann, braucht man erst einmal einen Wortschatz. Wie ist so ein Wortschatz aufgebaut? Die Liste in Tabelle 1 stellt einen solchen Relativen-

**Tabelle 1. Wortschatz (relative Datei)**

| Nr. | Wort | Code |
|---|---|---|
| 1 | abwaerts | 2010 |
| 2 | aufwaerts | 2009 |
| 3 | bedanken | 1074 |
| 4 | bedecken | 1037 |
| 5 | befestigen | 1042 |
| 6 | bekleiden | 1065 |
| 7 | belohnen | 1046 |
| 8 | benutzen | 1063 |
| 9 | berauben | 1058 |
| 10 | beruehren | 1017 |
| 11 | bestechen | 1068 |
| 12 | betatschen | 1017 |
| 13 | betrachten | 1011 |
| 14 | betreten | 1018 |
| 15 | betritt | 1018 |
| 16 | bewaessere | 1050 |
| 17 | bewegen | 1025 |
| 18 | bezahlen | 1046 |
| 19 | bieten | 1043 |
| 20 | binden | 1042 |
| 21 | bitten | 1066 |
| 22 | blasen | 1067 |
| 23 | blockieren | 1075 |
| 24 | brennen | 1045 |
| 25 | bruellen | 1070 |
| 26 | buchstabieren | 1016 |
| 27 | buddle | 1015 |
| 28 | danken | 1074 |
| 29 | dechiffrieren | 1038 |
| 30 | druecken | 1051 |
| 31 | entfachen | 1045 |
| 32 | entkommen | 1019 |
| 33 | entkorken | 1034 |
| 34 | entleeren | 1039 |
| 35 | entraetsle | 1038 |
| 36 | entriegle | 1034 |
| 37 | entziffre | 1038 |
| 38 | entzuenden | 1045 |
| 39 | erbitten | 1066 |
| 40 | ergreifen | 1012 |
| 41 | ermorden | 1057 |
| 42 | erschiessen | 1057 |
| 43 | erstechen | 1057 |
| 44 | erwerben | 1032 |
| 45 | erwirb | 1032 |
| 46 | erwuergen | 1057 |
| 47 | finden | 1041 |
| 48 | fliehen | 1019 |
| 49 | fragen | 1066 |
| 50 | fuellen | 1040 |
| 51 | geben | 1043 |
| 52 | gebrauchen | 1063 |
| 53 | gehen | 1001 |
| 54 | gellen | 1070 |
| 55 | gib | 1043 |
| 56 | giessen | 1050 |
| 57 | graben | 1015 |
| 58 | grabschen | 1017 |
| 59 | greifen | 1012 |
| 60 | haendigen | 1043 |
| 61 | heben | 1044 |
| 62 | hinab | 2010 |
| 63 | hinauf | 2009 |
| 64 | hinunter | 2010 |
| 65 | hoch | 2009 |
| 66 | holen | 1012 |
| 67 | huepfen | 1024 |
| 68 | informieren | 1069 |
| 69 | inventur | 1013 |
| 70 | kaufen | 1032 |
| 71 | kitzle | 1075 |
| 72 | klauen | 1058 |
| 73 | koepfen | 1057 |
| 74 | kontrollieren | 1063 |
| 75 | kuessen | 1071 |
| 76 | laufen | 1001 |
| 77 | leeren | 1039 |
| 78 | legen | 1014 |
| 79 | lesen | 1016 |
| 80 | lies | 1016 |
| 81 | liste | 1013 |
| 82 | n | 2001 |
| 83 | nehmen | 1012 |
| 84 | nimm | 1012 |
| 85 | no | 2006 |
| 86 | norden | 2001 |
| 87 | nordosten | 2006 |
| 88 | nordwesten | 2005 |
| 89 | nw | 2005 |
| 90 | o | 2004 |
| 91 | oeffnen | 1034 |
| 92 | osten | 2005 |
| 93 | paddle | 1026 |
| 94 | pennen | 1028 |
| 95 | pfeifen | 1067 |
| 96 | polieren | 1048 |
| 97 | probieren | 1059 |
| 98 | pruefen | 1033 |
| 99 | putzen | 1048 |
| 100 | quetschen | 1051 |
| 101 | ra | 2009 |
| 102 | rasten | 1027 |
| 103 | rauchen | 1079 |
| 104 | rauf | 2009 |
| 105 | reden | 1073 |
| 106 | reiben | 1017 |
| 107 | reinigen | 1048 |
| 108 | reisen | 1052 |
| 109 | reiten | 1029 |
| 110 | rennen | 1001 |
| 111 | ru | 2010 |
| 112 | rudere | 1026 |
| 113 | rufen | 1070 |
| 114 | ruhen | 1027 |
| 115 | runter | 2010 |
| 116 | s | 2002 |
| 117 | saeubere | 1048 |
| 118 | sagen | 1073 |
| 119 | saufen | 1078 |
| 120 | schauen | 1011 |
| 121 | schenken | 1043 |
| 122 | schicken | 1055 |
| 123 | schieben | 1051 |
| 124 | schlafen | 1028 |
| 125 | schliessen | 1035 |
| 126 | schmecken | 1059 |
| 127 | schreien | 1070 |
| 128 | schuettle | 1049 |
| 129 | schwimmen | 1021 |
| 130 | senden | 1055 |
| 131 | setzen | 1030 |
| 132 | so | 2008 |
| 133 | spielen | 1072 |
| 134 | sprechen | 1073 |
| 135 | sprich | 1073 |
| 136 | springen | 1024 |
| 137 | stecke | 1047 |
| 138 | stehen | 1031 |
| 139 | stehlen | 1058 |
| 140 | stoepsle | 1047 |
| 141 | strecken | 1053 |
| 142 | streichle | 1017 |
| 143 | suchen | 1041 |
| 144 | sueden | 2002 |
| 145 | suedosten | 2008 |
| 146 | suedwesten | 2007 |
| 147 | summen | 1067 |
| 148 | sw | 2007 |
| 149 | taetschle | 1017 |
| 150 | tauchen | 1022 |
| 151 | toeten | 1057 |
| 152 | tragen | 1065 |
| 153 | tragen | 1065 |
| 154 | trinken | 1078 |
| 155 | umhuellen | 1037 |
| 156 | unterrichten | 1069 |
| 157 | untersuchen | 1011 |
| 158 | verankere | 1042 |
| 159 | verbinden | 1042 |
| 160 | verbrennen | 1045 |
| 161 | verdecken | 1037 |
| 162 | verdrehen | 1061 |
| 163 | verhuellen | 1037 |
| 164 | verkaufe | 1054 |
| 165 | verknoten | 1042 |
| 166 | verkorken | 1036 |
| 167 | verlassen | 1020 |
| 168 | verlieren | 1014 |
| 169 | verlocken | 1068 |
| 170 | verloeschen | 1062 |
| 171 | vernichten | 1056 |
| 172 | verriegle | 1036 |
| 173 | verruecken | 1025 |
| 174 | verschieben | 1025 |
| 175 | verschliessen | 1036 |
| 176 | versperren | 1036 |
| 177 | versuchen | 1059 |
| 178 | w | 2003 |
| 179 | wandere | 1001 |
| 180 | warten | 1027 |
| 181 | wecken | 1076 |
| 182 | wedeln | 1064 |
| 183 | werfen | 1014 |
| 184 | westen | 2003 |
| 185 | winden | 1060 |
| 186 | winken | 1064 |
| 187 | wirf | 1014 |
| 188 | wuehlen | 1015 |
| 189 | zerquetschen | 1051 |
| 190 | zerreissen | 1052 |
| 191 | zerren | 1052 |
| 192 | zerschlagen | 1056 |
| 193 | zerschmettere | 1056 |
| 194 | zerstoeren | 1056 |
| 195 | zertruemmere | 1056 |
| 196 | ziehen | 1025 |
| 197 | zuenden | 1045 |

Datei-Wortschatz dar. Sie wurde mit Hilfe des Relativen-Dateien-Editors zusammengestellt und anschließend ausgedruckt.

Dieser Wortschatz besteht bis jetzt nur aus Verben und Himmelsrichtungsangaben. Diese Verben können als eine komplette Verbensammlung betrachtet werden, die man in einem Adventure benötigt. Das heißt, mit diesen Verben läßt sich nahezu jeder Befehlssatz ausdrücken (in Verbindung mit entsprechendem Objekt natürlich).

Wie liest man diese Wortschatztabelle? Stellen wir diese Frage erst einmal zurück! Nehmen Sie zunächst bitte einmal eine leere Diskette her und formatieren Sie diese. Dann legen Sie mit Hilfe des Relativen-Datei-Editors auf dieser Disk eine relative Datei mit folgenden Angaben an: 600 Sätze mit je 25 Zeichen. In diese relative Datei schreiben Sie dann die Wortschatz-Tabelle. Das geht einfach: Die linke Zahl in der Tabelle gibt jeweils die Satznummer der relativen Datei an. Danach kommt der Satzinhalt: Verbname und Codierungszahl. Bild 3 zeigt ganz genau, wie der erste Satz in der relativen Datei aussehen muß. Die Wörter (=Verben, Objekte etc.) dürfen also maximal 21 Zeichen lang sein. Die letzten vier Zeichen der Sätze sind für die Codierungszahl reserviert, deren Bedeutung uns im Moment noch nicht interessieren soll.

Beim »Abtippen« ist zu beachten, daß Sie nicht vom deutschen Zeichensatz Gebrauch machen. Das heißt, für »ae« darf nicht »ä« gesetzt werden und so weiter.

Wir haben jetzt also einen Wortschatz (bisher zwar nur Verben, aber das ist zunächst egal), auf den wir zurückgreifen können.

```
50000 REM ******************************      <227>
50001 REM *                            *      <009>
50002 REM *      WORT-PARSER 4.0       *      <169>
50003 REM *                            *      <011>
50004 REM *       <C> 1986 BEI         *      <086>
50005 REM *                            *      <013>
50006 REM *      MICHAEL NICKLES       *      <082>
50007 REM *                            *      <015>
50008 REM ******************************      <235>
50010 REM BEFEHLSSATZEINGABE -------------
      -------------------------------------
      --                                      <069>
50011 :                                       <202>
50015 SL=80: REM BEFEHLSSATZLAENGE            <144>
50020 PRINT"{DOWN}":BE$="":POKE 198,0:POKE
       211,0:POKE 214,22:SYS 58732:PRINT"{
      YELLOW}@";                              <156>
50030 GET X$:IF PEEK(203)=1 THEN 50120        <226>
50040 IF X$=""THEN 50030                      <171>
50050 IF LEN(BE$)=0 AND ASC(X$)=20 THEN 50
      030                                     <148>
50060 I=ASC(X$):IF I<32 OR I>133 AND I<159
       THEN IF I<>20 THEN 50030               <131>
50070 IF LEN(BE$)=SL AND I<>20 THEN 50030     <128>
50080 BE$=BE$+X$                              <031>
50090 PRINT CHR$(20);X$;"@";                  <108>
50100 IF I=20 THEN BE$=LEFT$(BE$,LEN(BE$)-
      2):GOTO 50030                           <101>
50110 GOTO 50030                              <001>
50120 PRINT"{LEFT,SPACE}"                     <227>
```

**Listing 1. Befehlssatzeingabe**

```
51000 REM BEFEHLSZERLEGUNG-CODIERUNG -----
      -------------------------------------
      --                                      <174>
51001 :                                       <176>
51010 VG$="NIMM":VG=1012                      <081>
51020 REM SUCHEN OB VG$ IN BE$ VORKOMMT       <057>
51030 FOR I=1 TO LEN(BE$)-LEN(VG$)+1          <010>
51040 : IF VG$=MID$(BE$,I,LEN(VG$)) THEN V
      E=VG                                    <088>
51050 NEXT I                                  <077>
51060 IF VE=0 THEN PRINT VG$" KOMMT NICHT
      IN BE$ VOR"                             <092>
51070 IF VE=1012 THEN PRINT VG$" KOMMT IN
      BE$ VOR"                                <212>
```

**Listing 2. Befehlszerlegung**

In der Einleitung zum Kapitel Parsertechnik habe ich Ihnen bereits den Grundaufbau eines JEDEN Parsers vorgestellt. Am Anfang jedes Parsers kommt folglich die Befehlseingabe, die beim C64 in der Regel aus einer modifizierten INPUT-Routine besteht.

Nach Durchlauf der Befehlseingabe liegt der Befehl beziehungsweise Befehlssatz als String BE$ vor. Lassen Sie mich nun wieder einmal von meinem alten Lieblingsbefehlssatz ausgehen: NIMM DAS SCHWERT. Also BE$= »NIMM DAS SCHWERT«. Dieser Befehlssatz soll jetzt anhand Vergleich mit dem Wortschatz codiert werden. Das heißt, dem VERB im Satz wird eine Verbzahl VE zugeordnet, dem Objekt eine Objektzahl OB, unbedeutende Wörter (zum Beispiel der, die, das und so weiter) werden ignoriert. Es gibt nun verschiedene Möglichkeiten, einen String beziehungsweise Befehlssatz zu codieren, nach Wörtern zu durchsuchen. Diese Möglichkeiten, ihre Vor- und Nachteile möchte ich Ihnen nun im folgenden einmal ganz genau vorstellen:

## Möglichkeit 1

Bitte tippen Sie Listing 1 ab und ergänzen Sie es um Listing 2.

Bei dieser Befehlszerlegungs-Methode wird der Befehlssatz eigentlich gar nicht zerlegt. Es wird einfach überprüft, ob das Verb »NIMM« (stellvertretend für die GANZE Worttabelle) im Befehlssatz BE$ enthalten ist. Dazu wird der gesamte Befehlsstring durchlaufen und mittels MID$-Funktion überprüft, ob das Verb NIMM (=VG$) im String vorkommt. Bild 4 demonstriert diesen Suchvorgang für den Befehlssatz »NIMM DAS SCHWERT«: das heißt, nehmen wir lieber als Befehlssatz »OTTO BITTE NIMM ES«, damit es nicht zu einfach wird. Aus Bild 4 wird auch deutlich, warum es legitim ist, anstelle von LEN(BE$) LEN(BE$)-LEN(VG$)+1 zu setzen. Es ist schließlich logisch, daß ein Suchbegriff von 4 Zeichen Länge spätestens in den letzten 4 Zeichen von BE$ gefunden werden muß.

25 Zeichen
A B W A E R T S | 2 0 1 0
21 Zeichen für Wort — 4 Zeichen für Codierungszahl

**Bild 3. Satzaufbau in der relativen Datei**

```
VG$= NIMM
BE$= OTTO BITTE NIMM ES
NIMM
 NIMM
  NIMM
   NIMM
    NIMM
     NIMM
      NIMM
       NIMM
        NIMM
         NIMM
GEFUNDEN!→ NIMM
           NIMM
            NIMM
             NIMM
Suchvorgang:
FOR I=1 TO
LEN(BE$)-
LEN(VG$)+1
```

**Bild 4. Beispiel für einen Suchvorgang**

Achtung: Beim bisherigen Programm wird VG$ im gesamten BE$ gesucht. Es wird so zum Beispiel für BE$=»OTTO BITTE NIMM ES«, aber auch in »OTTOBITTENIMMES« gefunden. Will man nach absoluten Wörtern (zum Beispiel ELIZA-Schlüsselwörter) suchen, so muß sichergestellt sein, daß überprüft wird, ob sich am Ende und Anfang des gesuchten Wortes ein SPACE befindet.

Listing 2 muß für diesen Fall so modifiziert werden:

```
51015 BE$=BE$+" "+BE$+" "
51040 : IF " "+VG$+" "=MID$(BE$,I,LEN(VG$)+2) THEN
 VE=VG
```

Methode 1 ist ganz die Methode, die bei ELIZA angewandt wird – nämlich dann, wenn die Schlüsselwörter gesucht werden. Das heißt, alle 38 Schlüsselwörter der ELIZA werden mittels dieser Methode in BE$ gesucht. Nicht zuletzt aus diesem Grund braucht ELIZA ziemlich viel Zeit, bis sie zu einer Antwort kommt. Gehen wir nun einmal von einem Wortschatz für ein Adventure aus, der ca. 600 Wörter groß ist. Ich sagte oben ja bereits, daß das Verb NIMM stellvertretend für den ganzen Wortschatz gesucht wird. Tatsächlich müßten alle 600 Wörter im Befehlssatz gesucht werden. Dies würde viel zuviel Zeit in Anspruch nehmen und außerdem zu Schwierigkeiten führen, wenn sich ein Befehlssatz aus mehreren Einzelbefehlen, die durch UND oder KOMMA getrennt sind, zusammensetzt. Methode 1 ist im Bereich der Adventure-Parser-Programmierung folglich völlig unbrauchbar. Aber auch was die Anwendung bei ELIZA anbelangt, gibt es inzwischen eine wesentlich bessere Methode zum Suchen eines Strings VG$ in einem String BE$:

## Möglichkeit 2

Ergänzen Sie diesmal Listing 1 mit Listing 3. Geben Sie auch hier als Befehlssatz BE$ wieder »OTTO BITTE NIMM ES« ein. Auch bei dieser Routine handelt es sich um eine Routine, die den Befehlsstring BE$ durchläuft und VG$ (wieder »NIMM«, stellvertretend für den gesamten Wortschatz) darin sucht. Allerdings erfolgt das Suchen nicht wie bei Methode 1 in »ein Zeichen«-Schritten. Bild 5 verdeutlicht den Suchvorgang bei der hier verwendeten Routine.

Während beim gleichen Befehlssatz BE$ bei Methode 1 ganze 12 Schritte bis zum Finden von VG$ in BES benötigt wurden, sind es hier nur noch 4. Der in Methode 2 angewandte Algorithmus wurde übrigens erst vor kurzem erfunden und in der 64'er Ausgabe 2/85 im Kurs »Effektives Programmieren« beschrieben.

Wie funktioniert das in Bild 5? Nun, der Suchvorgang beginnt ganz klar am Anfang des Befehlssatzes. Besteht hier noch keine Übereinstimmung, so wird VGS einfach um eine bestimmte Zeichenanzahl nach rechts verschoben. Woher weiß das Programm, um wie viele Zeichen es bei Nichtübereinstimmung verschieben muß? Das Prinzip ist eigentlich ganz einfach: Zunächst wird eine Tabelle angelegt, aus der hervorgeht, wie viele Zeichen jeder Buchstabe im Suchwort VG$ entfernt ist. Bei unserem Suchwort VG$=»NIMM« gilt folgendes:

M = 0 entfernen. Bei doppelt auftretenden Buchstaben wie zum Beispiel M im Wort NIMM, wird der Buchstabe notiert, der dem Ende des Wortes näher ist. Für alle Buchstaben, die im Suchwort VG$ nicht enthalten sind, wird als »Entfernungswert« die Länge von VG$ eingetragen – also 4 bei VG$=»NIMM«.

**Für das Wort NIMM ergibt sich die folgende Tabelle:**

VG$ = NIMM

| | | | | | |
|---|---|---|---|---|---|
| A | = 4 | J | = 4 | S | = 4 |
| B | = 4 | K | = 4 | T | = 4 |
| C | = 4 | L | = 4 | U | = 4 |
| D | = 4 | M | = 0 | V | = 4 |
| E | = 4 | N | = 3 | W | = 4 |
| F | = 4 | O | = 4 | X | = 4 |
| G | = 4 | P | = 4 | Y | = 4 |
| H | = 4 | Q | = 4 | Z | = 4 |
| I | = 2 | R | = 4 | | |

Das Erstellen dieser Tabelle findet beim Programm in den Zeilen 51020-51052 statt. Aus dieser Tabelle kann nun immer ermittelt werden, um wie viele Zeichen VG$ nach einem mißglückten Versuch nach rechts verschoben werden darf. Und das geht so: Betrachten Sie bitte wieder Bild 5. VG$ wird von hinten nach vorn beziehungsweise von rechts nach links durchlaufen. Dabei wird jedes Zeichen mit der entsprechenden Stelle in BE$ verglichen. Stimmen alle Zeichen überein, so ist das Word VG$ in BE$ gefunden. Erfolgt jedoch bei einem Buchstabenvergleich keine Übereinstimmung, so darf BG$ nach rechts verschoben werden – und zwar genau um die Zeichenzahl beziehungsweise zu dem Wert, der gemäß der Tabelle dem Zeichen von BE$ zugeordnet ist, bei dem zum ersten Mal keine Übereinstimmung festgestellt wurde. Beim ersten Suchschritt wird NIMM mit OTTO verglichen. Es wird also schon beim letzten Buchstaben Unübereinstimmung festgestellt, daß M ungleich O ist. Gemäß der Tabelle hat O den Wert 4. VG$ darf somit um 4 Zeichen nach rechts verschoben werden. Allerdings spielt der Tabellenwert nicht die einzige Rolle beim Verschieben. Vom aus der Tabelle ermittelten Wert wird noch die Entfernung des Buchstabens vom Wortende abgezogen, bei der

```
VG$= N I M M

BE$= O T T O  B I T T E  N I M M  E S
     N I M M
              N I M M
                     N I M M
                          N I M M
```

Bild 5. So kann schneller gesucht werden

```
10 DIM M%(26)                                  <181>
51000 REM BEFEHLSZERLEGUNG-CODIERUNG -----
      ------------------------------------
      --                                       <174>
51001 :                                        <176>
51010 VG$="NIMM":VG=1012                       <081>
51020 REM M$-TABELLE ANLEGEN ----------        <219>
51040 FOR I=1 TO 26:M%(I)=LEN(VG$):NEXT I      <077>
51050 FOR I=1 TO LEN(VG$):M%(ASC(MID$(VG$,
      I,1))-64)=LEN(VG$)-I:NEXT I              <101>
51052 FOR I=1 TO LEN(VG$):PRINT MID$(VG$,I
      ,1),M%(ASC(MID$(VG$,I,1))-64):NEXT I     <158>
51060 REM VG$ IN BE$ SUCHEN -----------        <035>
51065 PRINT"{CLR,SPACE}";BE$                   <120>
51070 PO=1                                     <025>
51080 FOR I=1 TO PO:PRINT" ";:NEXT I:PRINT
       VG$                                     <104>
51085 IF VG$=MID$(BE$,PO,LEN(VG$))THEN PRI
      NT"GEFUNDEN !":END                       <214>
51090 FOR I=LEN(VG$)TO 1 STEP-1                <107>
51095 :IF MID$(BE$,PO+I,1)=""THEN PRINT"NI
      CHT GEFUNDEN!":END                       <231>
51105 :IF MID$(VG$,I,1)=MID$(BE$,PO+I-1,1)
      THEN 51120                               <218>
51109 :IF ASC(MID$(BE$,PO+I-1,1))=32 THEN
      PO=PO+LEN(VG$):I=0:GOTO 51120            <022>
51110 :PO=PO+M%(ASC(MID$(BE$,PO+I-1,1))-64
      )+LEN(VG$)-I:I=0                         <216>
51120 NEXT I                                   <147>
51130 GOTO 51080                               <087>
```

Listing 3. Zweite Möglichkeit zur Befehlszerlegung

die erste Nichtübereinstimmung gefunden wurde. Dies ist nicht ganz einfach verständlich. Ich rate Ihnen daher eine eigenständige Analyse von Listing 3, Zeile 51060-51130. Danach wird Ihnen das Prinzip dieser Methode mit Sicherheit klar sein.

Man kann nicht so ohne weiteres behaupten, daß die zweite Methode der ersten überlegen ist. Auf den ersten Blick scheint Methode 2 zwar schneller zu sein, da weniger Verschiebungen notwendig sind. Dem steht jedoch eine wesentlich größere Rechenzeit als bei Methode 1 gegenüber. Am besten, Sie probieren selbst einmal aus, mittels TI$-Zeitmessung, welche der beiden Methoden die geeignetere ist. Auf jeden Fall ist die Methode 2 äußerst interessant, nicht zuletzt deshalb, weil sie erst vor kurzem entdeckt wurde.

## Möglichkeit 3

Wie Methode 1 eignet sich Methode 2 jedoch kaum zur Adventureprogrammierung, sondern vielmehr zur Entwicklung von ELIZA-ähnlichen Simulationen. Anders Methode 3:

Die 3. Methode entspricht weitgehend der Methode, die ich bereits in meinem ersten Adventure-Kurs verwendet habe. Es geht darum, den eingegebenen Befehlssatz BE$ zunächst in einzelne Worte BE$(0)-BE$(N) zu zerlegen, wobei N für die Anzahl der im Befehlssatz vorkommenden Worte steht. Der Befehlssatz kann in BE$(0)="NIMM" BE$(1)="DAS" und BE$(2)="SCHWERT" zerlegt werden.

BE$(3) ist hier natürlich " ", wodurch das Satzende gekennzeichnet ist.

Nun wurde eine Wortzählvariable WZ auf 0 gesetzt. Sodann wurde das Wort BE$(WZ) codiert – also im Wortschatz gesucht. Es wird so eine Verbzahl beziehungsweise Objektzahl etc. ermittelt. Wird das Wort nicht im Wortschatz gefunden, so kommt die Fehlermeldung »ICH KENNE DAS WORT...NICHT«, und der gesamte Codiervorgang wurde abgebrochen. Sodann wurde der Wortzähler WZ um 1 erhöht, und eine neue Suchrunde begann. Der ganze Satz wurde so also codiert. Der Suchvorgang wurde dann abgeschlossen, wenn BE$(WZ)=» « war (also Satzende) oder BE$(WZ)=»UND« beziehungsweise »,« war (=Befehlssatz mit mehreren Befehlen). Wurde der Suchvorgang wegen letzterem unterbrochen, so wurde die Variable UD auf den Wert 1 gesetzt. Dadurch wußte das Programm, daß beim nächsten Mal, wenn der Spieler dran ist, die Befehlseingabe (INPUT-ROUTINE) übersprungen werden muß und die Codierung im alten Befehlssatz dort fortgesetzt werden mußte, wo man zuletzt wegen einem UND beziehungsweise KOMMA abgebrochen hatte.

In Bild 6a und 6b finden Sie nochmals die beiden Ablaufschemen, die diese Codiermethode beschreiben:

Wer meinen ersten Adventurekurs durchgearbeitet hat, dem sind diese beiden Ablaufschemen mit Sicherheit gut bekannt.

Obwohl die Ablaufschemen bereits über 1 Jahr alt sind und es inzwischen gravierende Verbesserungen im Bereich der Adventureprogrammierung gegeben hat (wie Sie schon bald

GESAMTABLAUFSCHEMA
TITELBILD D. ADVENTURES
SPIELANLEITUNG
TABELLEN DEFINIEREN ETC.
BEFEHLSEINGABE
BEFEHLSCODIERUNG
IST UD=1 ?
JA
NEIN
REAKTION AUF BEFEHL
ACTION
VERLOREN
ODER GEWONNEN ?
NEIN
JA
SPIELENDE

Bild 6a. Aufbau und Ablauf eines Spiels

ANFANG-CODIERVORGANG
IST UD=1 ?
JA
NEIN
UD=0
WZ=0:VE=0:OB=0
BE$(WZ) GEMÄSS
DER WORTSCHATZTAB.
CODIEREN
WZ=WZ+1
IST BE$(WZ)="UND"
JA
NEIN
UD=1
IST BE$(WZ)=" "
NEIN
JA
ENDE-CODIERVORGANG

Bild 6b. Die Befehlscodierung im Detail

sehen werden !), haben sie keineswegs ihre Gültigkeit verloren.

Bild 6a stellt das Gesamtablaufschema eine Adventures dar. Sie können jedes Adventure-Programmierkonzept in dieses Schema quetschen. Unter Adventure läßt sich natürlich auch jedes andere KI-Programm verstehen, bei dem im Programm folgende Teile enthalten sind: Befehlseingaberoutine, Analyse des Befehlssatzes anhand eines gegebenen Wortschatzes, Interpretation der Analyse beziehungsweise Reaktion auf die Befehlseingabe.

Bild 6b beschreibt den Ablauf der Befehlscodierung nach Methode 3. Lassen Sie uns jetzt Methode 3 programmieren.

Ergänzen Sie dieses Mal Listing 1 bitte um Listing 4. Unser Parsersystem 4.0 ist nun insofern komplett, daß nach der Befehlseingabe der Befehlssatz BE$ in einzelne Worte BE$(1)–BE$(N) zerlegt wird. Dabei ist festgelegt, daß der Befehlssatz aus maximal 10 einzelnen Wörtern bestehen darf. Dies läßt sich natürlich leicht abändern, aber 10 sollen uns zunächst genug sein zum Experimentieren. Die Wortzerlegung funktioniert äußerst primitiv: Es wird einfach BE$ durchlaufen und nach den SPACEs zwischen den einzelnen Wörtern gesucht. Eine Dokumentation zu Listing 4 erübrigt sich (hoffe ich doch !!!).

Jetzt muß der Codiervorgang gemäß Bild 6b programmiert werden. Ergänzen Sie das jetzige Programm bitte um Listing 5.

Auch Listing 5 läßt sich leicht verstehen. Auf Einbau von UND-Befehlstrennungen habe ich der Einfachheit halber zunächst verzichtet. Listing 5 bringt im Moment allerdings noch gar nichts. Wir müssen uns zunächst noch überlegen, wie wir das Wort in unserer relativen Wortschatzdatei finden beziehungsweise die einzelnen Befehlswörter BE$(1)–BE$(N). Im letzten Adventurekurs war dies noch ganz einfach: Da gab es für Verben und Objekte einzelne Tabellen (dimensionierte Stringfelder), in denen gesucht wurde. Jetzt befindet sich der gesamte Wortschatz, also Verben, Objekte, unwichtige Wörter, Richtungen etc; in einer relativen Datei, die alphabetisch sortiert ist. Nun, wie findet man Wörter in einer sortierten relativen Datei am schnellsten ? Ganz klar – binäre Suchmethode. Wir müssen also in Listing 5 eine binäre Suchroutine einbauen. Eine solche Routine haben wir ja bereits im Kursabschnitt »Dateiverwaltung« kennengelernt. Bevor wir die binäre Suchroutine in Listing 5 einbauen, müssen wir uns allerdings noch einige Gedanken zur Organisation des Wortschatzes machen.

## Kleine Wortschatzlehre

Werfen Sie an dieser Stelle bitte nochmals einen Blick auf Bild 3. Dort habe ich Ihnen gezeigt, wie sich die 25 Zeichen eines jeden Satzes der Wortschatz-Datei aufteilen: Die ersten 21 Zeichen sind für das Wort selbst reserviert. Also zum Beispiel für NIMM, ÖFFNE (Verben) oder auch HAUSTÜR, SCHRANK (Objekte). Die letzten 4 Zeichen jedes Satzes sind für eine Zahl - die Codierungszahl - reserviert. Was hat es nun mit dieser Codierungszahl auf sich ? Ganz einfach, aus ihr läßt sich ermitteln, welcher Wortart das entsprechende Wort angehört und welche Zahl ihm als Codierung (zum Beispiel Verbzahl VE usw.) zugedacht ist. Die 4stellige Zahl ist also vielmehr als zwei Zahlen beziehungsweise eine Ziffer und eine Zahl zu verstehen, die unmittelbar nebeneinander stehen. Bild 7 macht die genaue Unterteilung deutlich:

Die erste Ziffer der 4stelligen Zahl beschreibt somit die Wortart WA, während die letzten drei Ziffern den Wortcode WC bestimmen. Wir können also festlegen:

Wortart: WA= VAL(MID$(A$,22,1))

Wortcode: WC= VAL(MID$(A$,23,3))

A$ steht dabei für einen beliebigen Satz der Wortschatz-Datei. Das tatsächliche Wort ermittelt sich als

A$=LEFT$(A$,21)

Allerdings hat das absolute Wort nicht immer die Länge von 21 Zeichen. Es wäre jedoch günstig, wenn in A$ nur das Wort (keine SPACEs mehr danach) stehen würde. Die SPACEs kann man ganz einfach eliminieren:

```
9340 IFRIGHT$(A$,1)="cursor right" THENA$=LEFT$
(A$,LEN(A$-1)) :GOTO 9340
```

Diese Zeile wiederholt sich so lange, bis in A$ nur noch das Wort und keine SPACEs mehr danach stehen. Eigentlich darf ich hier gar nicht von SPACEs reden. Was wir auf dem Bildschirm als SPACE sehen, ist bei Sätzen, die von relativen Dateien, die mit unserem Relative-Dateien-Editor erstellt wurden, stammen, tatsächlich ein »cursor right«-Zeichen. Achten Sie also stets darauf, daß Sie in Sätzen aus unseren Dateien niemals nach einem SPACE, sondern vielmehr nach dem cursor-right-Zeichen suchen (vergleiche Beschreibung zum Relative-Dateien-Editor im Kapitel »relative Dateien«)!

Natürlich muß jetzt noch festgelegt werden, welche Wortartzahl welcher Wortart entspricht:

1 = Verb
2 = Richtung
3 = Objekt
4 = Aus-Wort

Bisher treten in unserem Wortschatz nur 1 und 2 als WA-Werte auf – kein Wunder, denn unser Ausgangs-Wortschatz besteht ja bisher nur aus Verben und Richtungen (die im Prinzip auch als Verben verstanden werden können, jedoch trotzdem einen eigenen WA-Wert benötigen, wie Sie gleich sehen werden).

Unter Aus-Wort verstehe ich jedes Wort, das im Befehlssatz auftreten kann, bei der Codierung jedoch keinerlei Rolle spielt. Gemeint sind Wörter wie zum Beispiel der, die, das, nach, vom und so weiter.

```
51000 REM BEFEHLSZERLEGUNG-CODIERUNG -----
      --------------------------------------
      --                                  <174>
51001 :                                   <176>
51005 REM BE$ IN BE$(1)-BE$(N) ZERLEGEN   <165>
51010 FOR I=0 TO 10:BE$(I)="":NEXT I      <183>
51020 WZ=0                                <059>
51030 FOR I=1 TO LEN(BE$)                 <004>
51040 : IF MID$(BE$,I,1)=" "THEN WZ=WZ+1:G
      OTO 51060                           <223>
51045 : IF WZ>10 THEN PRINT"EINGABE IST ZU
       LANG !":I=LEN(BE$)+1:GOTO 51060    <037>
51050 : BE$(WZ)=BE$(WZ)+MID$(BE$,I,1)     <247>
51060 NEXT I                              <087>
```

**Listing 4. Befehlscodierung nach Methode 3**

```
51100 REM BE$(1)-BE$(N) CODIEREN          <107>
51110 WZ=0      :REM WORTZAEHLER AUF 0    <094>
51120 VE=0:OB=0:REM ALTE ERG. LOESCHEN    <247>
51130 REM WORT BE$(WZ) IM WORTSCHATZ SUCHE
      N                                   <065>
51140 :                                   <059>
51150 :                                   <069>
51160 :                                   <079>
51170 :                                   <089>
51180 :                                   <099>
51190 :                                   <109>
51200 WZ=WZ+1                             <241>
51220 IF BE$(WZ)=""THEN 52000             <197>
51230 GOTO 51130                          <155>
52000 REM REAKTION AUF BEFEHL ------------
      --------------------------------------
      --                                  <118>
```

**Listing 5. Der Codiervorgang nach Bild 6b**

| A | B | W | A | E | R | T | S | | | | | | | | | | | | | | 2 | 0 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|

WORT — WORTART — WORTCODE

**Bild 7. Die Wortcodierung im Datensatz**

```
10 OPEN 1,8,3,"WORTSCHATZ":OPEN 15,8,15:RE
   M WORTSCHATZ-DATEI OEFFNEN                <125>
20 INPUT SU$                                 <157>
30 GOTO 9500                                 <114>
9000 REM ZEIGER POSITIONIEREN -------------
     --------------------------------------
     -                                       <187>
9001 :                                       <087>
9010 HB%=SA/256:LB%=SA-HB%*256               <068>
9020 PRINT#15,"P";CHR$(3);CHR$(LB%);CHR$(H
     B%);CHR$(1)                             <005>
9030 INPUT#15,A:IF A=50 THEN PRINT"SATZNUM
     MER ZU HOCH"                            <088>
9040 RETURN                                  <208>
9100 REM SATZ LESEN -----------------------
     --------------------------------------
     -                                       <178>
9101 :                                       <187>
9110 IF A=50 THEN RETURN                     <082>
9120 INPUT#1,A$                              <116>
9130 RETURN                                  <042>
9200 REM SATZ SCHREIBEN -------------------
     --------------------------------------
     -                                       <105>
9201 :                                       <031>
9210 IF A=50 THEN RETURN                     <182>
9220 PRINT#1,A$                              <042>
9230 RETURN                                  <144>
9300 REM SATZ-TEILBEREICH LESEN -----------
     --------------------------------------
     -                                       <021>
9301 :                                       <133>
9310 IF A=50 THEN RETURN                     <028>
9320 INPUT#1,A$                              <062>
9330 WA=VAL(MID$(A$,22,1)):REM WORTART       <075>
9332 WC=VAL(MID$(A$,23,3)):REM WORTCODE      <100>
9335 A$=LEFT$(A$,21)        :REM WORT        <075>
9340 IF RIGHT$(A$,1)="{RIGHT}"THEN A$=LEFT
     $(A$,LEN(A$)-1):GOTO 9340               <171>
9345 PRINT A$,LEN(A$)                        <049>
9350 RETURN                                  <008>
9500 REM BINAERE SUCHROUTINE -------------
     --------------------------------------
     -                                       <166>
9520 REM SU$=SUCHWORT                        <044>
9540 SZ=197:N=INT(LOG(SZ)/LOG(2))+1
                    :REM MAXIMAL-FORMEL      <224>
9560 SA=(2↑N)/2:REM MITTE DER GES.DAT        <209>
9565 GOSUB 9000:GOSUB 9300
                     :REM A$ LESEN           <121>
9580 N=N-1      :REM 1.ABFRAGE               <234>
9600 :                                       <178>
9620 REM SU$ MIT A$ VERGLEICHEN ------       <251>
9623 : IF LEN(SU$)<3 THEN 9635               <189>
9625 : IF WA=1 AND SU$=LEFT$(A$,LEN(SU$))
     THEN PRINT"SU$=VERB:SATZ ";SA:END       <223>
9630 : IF WA=3 AND SU$=RIGHT$(A$,LEN(SU$))
     THEN PRINT"SU$=OBJEKT:SATZ ";SA:END     <172>
9635 : IF WA=2 AND SU$=A$
     THEN PRINT"SU$=RICHTUNG:SATZ ";SA:END <192>
9637 : IF WA=1 AND SU$=A$
     THEN PRINT"SU$=VERB:SATZ ";SA:END       <094>
9640 : IF SU$<A$THEN SA=SA-(2↑(N-1)):GOSUB
      9000:GOSUB 9300:GOTO 9700              <060>
9660 : IF SA+(2↑(N-1))>SZ AND N>-1 THEN N=
     N-1:GOTO 9660        :REM ''UEBERLAUF'' <038>
9680 : SA=SA+(2↑(N-1)):GOSUB 9000:GOSUB 93
     00               :REM ''HALBIEREN''     <154>
9700 :                                       <022>
9720 :N=N-1     :REM NAECHST. HALBIEREN      <021>
9740 :IF N<0 THEN PRINT"SU$ GIBT'S NICHT!
     IN 1220":END     :REM SU$ NICHT IN DAT
     EI                                      <043>
9760 PRINT"SA=";SA;" N=";N
                 :REM UNNOETIG !!            <199>
9780 :GOTO 9600:REM NAECHST. SUCH-VERS.      <069>
```

**Listing 6. Die Binäre Suchmethode**

| NIMM | | |
|---|---|---|
| SCHUETTLE | POLIEREN | NORDWESTEN |
| SA=128 N=8 | SA=96 N=5 | SA=88 N=3 |
| HINUNTER | LIES | NIMM |
| SA=64 N=6 | SA=80 N=4 | SA=84 N=2 |

**Bild 8. Binär gesucht = schnell gefunden**

Bei dem Wortschatz, den wir im letzten Adventurekurs verwendet haben, mußten die Wörter nicht mit Zahlen gekennzeichnet werden. Auch Wörter mit gleicher Bedeutung wurden anders abgehandelt. Die nun angewandte Methode hat natürlich ihre Vorteile: Zum einen läßt sich der Wortschatz leichter um Worte erweitern, und zum anderen müssen nicht mehr mehrere Einzeltabellen verwaltet werden (für die verschiedenen Wortarten).

Eine gute Wortschatzverwaltung beeinflußt nicht zuletzt auch die Anwenderfreundlichkeit eines Parsers in positivem Maße. Ein anwendungsfreundlicher Parser ist ein Parser, der bei der Befehlssatz-Analyse äußerst tolerant vorgeht.

Es muß zum Beispiel anstelle des Verbs NIMM auch NEHME und NEHM sowie gegebenenfalls auch HOLE und ERGREIFE verstehen. Der Trick bei dieser Sache ist, daß mehr Worte verstanden werden, als eigentlich im Wortschatz stehen. Was das soll? Ganz einfach:

Nehmen wir einmal an, wir wollen in einem Adventure die NIMM-Routine programmieren:

```
IF VERB=NIMM AND OBJEKT X  IST  HIER THEN PRINT
"OK. ICH HABE OBJEKT X."
```

So könnte eine vereinfachte Syntax lauten. Dem Verb NIMM ist gemäß der Wortschatztabelle der Wortwert 12 zugeordnet (also VE=12). Wir können also sagen:

```
IF VE=12 AND OBJEKT X IST HIER THEN PRINT "OK.
ICH HABE OBJEKT X".
```

Was passiert, wenn der Spieler den Befehlssatz »HOLE OBJEKT X« eingibt? Nun, das Programm antwortet mit »OK. ICH....«, da dem Verb HOLE ebenfalls der Wortwert (beziehungsweise Verbwert) 12 zugeordnet ist. Unser Wortschatz ist ziemlich intelligent aufgebaut – intelligent insofern, daß Verben gleicher Bedeutung der gleiche Verbwert zugeordnet ist. Ein guter Parser sollte allerdings »NIMM DAS SCHWERT« genauso gut verstehen wie »ICH WILL DAS SCHWERT NEHMEN« und »NEHM DAS SCHWERT« (auch wenn es grammatikalisch nicht in Ordnung ist). Wie stellt man es nun an, daß NEHM genauso gut verstanden wird wie NEHMEN? Nun, ganz einfach. NEHM steckt schließlich in NEHMEN drin. Nicht zuletzt ist dies auch der Grund, warum im Wortschatz immer die längstmögliche Form eines Verbs gespeichert wird – insofern ist dies möglich. Die meisten routinierten Adventure-Spieler geben eh in der Sytax NIMM, VERLIER, GIB etc. ein. Wir können uns deshalb auch hauptsächlich an dieser Form orientieren. Nehmen wir einmal an, der Satz »NIMM DAS SCHWERT« soll codiert werden (welcher sonst). BE$(0) ist dann ganz klar »NIMM«. Wir wollen BE$(0) jetzt anhand des Wortschatzes codieren. Mittels der binären Suchmethode werden dann verschiedene Sätze vom Wortschatz eingelesen (als String A$) und mit SU$(=BE$(0)) verglichen. Bild 8 zeigt, wie das Wort NIMM im Wortschatz nach der binären Suchmethode gefunden wird.

Wie wird A$ eingentlich mit SU$ verglichen?

Ein Vergleich IF A$=SU$ THEN PRINT »GEFUNDEN« wäre mit Sicherheit falsch, da man in diesem Fall niemals das Verb NEHM finden würde (im Wortschatz steht nur NEHME). Der korrekte Vergleich lautet daher:

```
IF SU$=LEFT$(A$,LEN(SU$)) THEN ? "GEFUNDEN"
```

Dadurch sind bei der Befehlseingabe auch Abkürzungen der Verben erlaubt – zum Beispiel UNTERSU statt UNTERSUCHEN beziehungsweise NEHM statt NEHMEN. Dies ist mit Sicherheit jedem sonnenklar. Was aber, wenn der Spieler nach Norden gehen will? Er wird dann mit größter Wahrscheinlichkeit BE$=»N« eingeben. Aufgrund unserer bisherigen Abfragemethode IF SU$=LEFT$(A$,...) wird das Programm annehmen, daß das erste Wort, das mit N beginnt, das richtige ist. Werfen Sie doch bitte einmal einen Blick in die Wortschatztabelle: Sie werden merken, daß dort für Norden sowohl »N« als auch »NORDEN« vermerkt ist. Nicht umsonst haben Richtungen deshalb einen anderen WA-Wert als Ver-

ben. Sicher erinnern Sie sich daran, daß A$ vor dem Vergleich mit SU$ präpariert wird. Zunächst wird A$ mit einem Satz der Wortschatzdatei gefüllt. Dann werden Wortcode WC und Wortart WA ermittelt. Zuletzt steht in A$ dann nur noch das absolute Wort. Dadurch, daß die Wortart beim Vergleichen bereits bekannt ist, kann folgender Vergleich programmiert werden:

```
IF WA=1 AND SU$=LEFT$(A$,LEN(SU$)) THEN ? "VERB GE-
FUNDEN"
IF WA=2 AND SU$=A$ THEN ? "RICHTUNG GEFUNDEN"
```

Sie sehen also, bei RICHTUNGEN muß SU$ mit A$ übereinstimmen und SU$ nicht nur linksbündig in A$ enthalten sein.

Wie muß nach Objekten gesucht werden? Wir wollen uns auch mit dieser Frage beschäftigen – auch wenn sich im Wortschatz bisher noch keine Objekte befinden.

Objektnamen setzen sich oft aus Hauptwörtern zusammen. So zum Beispiel HAUSTÜR, HOLZKISTE, WEINFLASCHE und so weiter. Will der Spieler eine Haustür öffnen, so kann er sagen »ÖFFNE DIE HAUSTÜR« oder auch »ÖFFNE DIE TÜR«. Der Parser muß also wissen, daß HAUSTÜR=TÜR, HOLZKISTE=KISTE und so weiter ist. Entscheidend zur Bestimmung eines Objektes ist stets das zweite Hauptwort (beziehungsweise das Hauptwort, das ganz rechts im Objektnamen steht, beziehungsweise das Objekt selbst, wenn es sich nicht aus mehreren Hauptwörtern zusammensetzt). Es bietet sich also an, bei der Objektsuche von »rechts her« zu vergleichen:

```
IF WA=3 AND SU$=RIGHT$(A$,LEN(SU$)) THEN ? "OBJEKT-
 GEFUNDEN"
```

Das war wieder mal ziemlich viel Theorie. Listing 6 setzt das eben Gelernte in die Praxis um.

**Dokumentation:**

| | |
|---|---|
| 10-30 | Hier wird die Wortschatzdatei geöffnet. Vor Programmstart ist es also erforderlich, daß Sie die Diskette mit der Wortschatzdatei einlegen. Anschließend wird ein Suchbegriff SUS eingegeben. |
| 9000-9230 | Hier stehen die drei Unterprogramme, die beim Arbeiten mit relativen Dateien immer wieder benötigt werden: Positionieren, Lesen und Schreiben. |
| 9300-9350 | Hier wird ein Satz A$ der Wortschatzdatei gelesen. Zunächst werden Wortcode WC und Wortart WA gelesen, dann die Codierzahl vom A$ abgeschnitten und alle SPACES (cursor-rights) hinter dem absoluten Wort weggenommen (Zeile 9340). |
| 9500-9780 | Hier steht die binäre Suchroutine, wie ich sie Ihnen bereits im Kapitel Dateiverwaltung Listing 15 beschrieben habe. Die Routine wurde zwecks Zusammenarbeit mit relativer Datei entsprechend modifiziert. |
| 9540 | Für SZ muß die Anzahl der Worte, die im Wortschatz enthalten sind, eingesetzt werden. (Im Moment haben wir 197 Worte.) |
| 9620-9637 | Hier werden die Objekte, Verben und Richtungen gesucht. |

```
0 REM ********************************** <131>
1 REM *   INDEX-MAKER - ERSTELLT DEN   * <097>
2 REM *   INDEX ZU EINEM BELIEBIGEN,   * <058>
3 REM *    ALPHABETISCH GEORDNETEN     * <134>
4 REM * BEREICH EINER RELATIVEN DATEI * <198>
5 REM * <C> 1986 BEI MICHAEL NICKLES   * <163>
6 REM ********************************** <137>
10 PRINT"{CLR,YELLOW,RVSON}INDEX-MAKER{RVO
   FF}"                                  <022>
20 DIM M%(26) : REM INDEX-TABELLE        <034>
30 PRINT"{DOWN}BITTE GEBEN SIE DEN NAMEN D
   ER RELATIVEN                          <039>
40 PRINT"DATEI AN, VON DER EIN INDEX ERSTE
   LLT                                   <250>
50 PRINT"WERDEN SOLL!"                   <202>
60 POKE 198,0:INPUT"{DOWN}";D$           <225>
70 OPEN 1,8,3,D$:OPEN 15,8,15:REM DATEI OE
   FFNEN                                 <197>
80 PRINT"{DOWN}BITTE GEBEN SIE DEN BEREICH
    DER DATEI AN";                       <143>
90 PRINT"VON DEM DER INDEX ERSTELLT WERDEN
    SOLL!"                               <061>
100 POKE 198,0:INPUT"{DOWN}ANFANG";AN    <133>
110 POKE 198,0:INPUT"{DOWN}ENDE{2SPACE}";E
    N                                    <086>
120 SA=AN:GOSUB 9000:IF A=50 THEN PRINT"AN
    FANGSATZ EXISTIERT NICHT!":GOTO 100  <236>
130 SA=EN:GOSUB 9000:IF A=50 THEN PRINT"EN
    DSATZ EXISTIERT NICHT!":GOTO 100     <190>
140 IF AN>EN THEN PRINT"UNSINN!":GOTO 100 <244>
1000 REM INDEX-ERSTELLEN ----------------
     ------------------------------------
     -                                   <138>
1001 :                                   <215>
1005 PRINT"{DOWN}BITTE WARTEN !"         <016>
1010 SA=AN:GOSUB 9000                    <149>
1020 IZ=0                                <042>
1030 FOR SA=AN TO EN                     <246>
1035 :                                   <251>
1040 : GOSUB 9100:PRINT SA,A$            <000>
1050 : IF M%(ASC(LEFT$(A$,1))-64)=0 THEN M
     %(ASC(LEFT$(A$,1))-64)=SA           <227>
1080 :                                   <040>
1100 NEXT SA                             <012>
2000 REM INDEX AUSGEBEN -----------------
     ------------------------------------
     -                                   <185>
2001 :                                   <199>
2010 PRINT"{CLR}INDEX-AUSGABE:"          <030>
2020 PRINT"{2DOWN}AUSGABE AUF BILDSCHIRM <
     1>                                  <068>
2030 PRINT"{DOWN}ALS SEQ-FILE AUF DISK{2SP
     ACE}<2>                             <247>
2040 POKE 198,0                          <170>
2050 GET I$:IF I$<>"1"AND I$<>"2"THEN 2050 <094>
2060 IF I$="1"THEN 2100                  <023>
2070 IF I$="2"THEN 2200                  <170>
2100 REM AUSGABE AUF BILDSCHIRM          <197>
2101 :                                   <045>
2110 PRINT"{CLR}"                        <066>
2120 FOR I=1 TO 13:PRINT CHR$(I+64)"="M%(I
     ),CHR$(I+13+64)"="M%(I+13):NEXT I   <086>
2130 END                                 <100>
2200 REM M%(1)-M%(26) ALS SEQFILE SPEICHER
     N                                   <168>
2201 :                                   <145>
2210 POKE 198,0:INPUT"{DOWN}FILENAME ";I$:
     I$=I$+",S,W"                        <043>
2220 OPEN 2,8,2,I$                       <006>
2230 FOR I=1 TO 26                       <223>
2240 :PRINT#2,M%(I)                      <103>
2250 NEXT I:CLOSE 2                      <113>
2260 END                                 <230>
9000 REM ZEIGER POSITIONIEREN -----------
     ------------------------------------
     -                                   <187>
9001 :                                   <087>
9010 HB%=SA/256:LB%=SA-HB%*256           <068>
9020 PRINT#15,"P";CHR$(3);CHR$(LB%);CHR$(H
     B%);CHR$(1)                         <005>
9030 INPUT#15,A:IF A=50 THEN PRINT"SATZNUM
     MER ZU HOCH"                        <088>
9040 RETURN                              <208>
9100 REM SATZ LESEN ---------------------
     ------------------------------------
     -                                   <178>
9101 :                                   <187>
9110 IF A=50 THEN RETURN                 <082>
9120 INPUT#1,A$                          <116>
9130 RETURN                              <042>
```

**Listing 7. Mit diesem Programm kann der Index einer relativen Datei erstellt werden**

9623 Wenn die Zeichenanzahl des Suchbegriffs weniger als 3 Zeichen beträgt, muß davon ausgegangen werden, daß es sich bei SU$ um eine Richtungsangabe handelt. Das heißt auch, daß Verbenabkürzungen mindestens 3 Buchstaben betragen müssen. Man kann für INVENTUR also nicht die Abkürzung I eingeben– es sei denn, I wurde zusätzlich im Wortschatz definiert (Zeile 9637 sorgt dafür, daß solche Verbabkürzungen akzeptiert werden).
Ist LEN(SU$) also kleiner als 3, so werden die Zeilen 9625 (Verbvergleich linksbündig) und Zeile 9630 (Objektvergleich rechtsbündig) übersprungen.

9625 Verbsuche (linksbündig)
9630 Objektsuche (rechtsbündig)
9635 Richtungssuche (absolut)
9637 Verbsuche (absolut) – für Verben, deren Abkürzung auch weniger als 3 Zeichen betragen darf – müssen im Wortschatz extra verzeichnet werden!

Zum Abschluß von Methode 3 müßte Listing 6 nun eigentlich mit Listing 5 kombiniert werden. Ich möchte jedoch davon absehen, da Sie dies zum einen leicht selbst machen können und es zum anderen noch eine bessere Möglichkeit gibt, die im Prinzip eine Verbesserung des letzten Programms bedeutet:

## Möglichkeit 4

Unser Wortschatz besteht aus 197 Worten. Um ein bestimmtes Wort mittels der binären Suchmethode in diesem Wortschatz zu finden, beziehungsweise um festzustellen, daß dieses Wort im Wortschatz nicht vorkommt, sind maximal 8 Lesezugriffe auf Disk erforderlich. Jeder Schreib- beziehungsweise Lesezugriff auf Disk kostet Zeit. Auch durch Assemblerprogrammierung läßt sich diese Zugriffszeit kaum verkürzen. Es gibt allerdings eine Suchmethode, bei der weniger Lesezugriffe aus Disk nötig sind als bei Möglichkeit 3.

## Das Index-Binäre-Suchprinzip

Keine Angst, dieses Prinzip läßt sich einfacher verstehen, als der komplexe Name vermuten läßt. Beim Index-Binären-Suchprinzip handelt es sich lediglich um eine Verbesserung der binären Suchmethode. Gehen wir einmal davon aus, daß unser Suchbegriff SU$=»NIMM« ist. Unser Parser erkennt nicht auf Anhieb, daß SU$ ein VERB ist. Er erkennt SU$ erst dann als Verb an, wenn er es im Wortschatz gefunden hat. Etwas könnte der Parser allerdings schon vor Beginn des Suchvorgangs wissen – und zwar, daß der Suchbegriff mit dem Buchstaben N beginnt. Wozu also den gesamten Wortschatz durchsuchen, der ja eh alphabetisch geordnet ist? Es würde doch vollkommen ausreichen, den Bereich, in dem alle mit N beginnenden Wörter stehen, mittels der binären Suchmethode zu durchsuchen! Dazu benötigt man zunächst einen Index – eine Tabelle, in der steht, ab welchem Satz die mit A beginnenden Wörter stehen, ab welchem die mit B beginnenden, und so weiter. Das Programm in Listing 7 ist in der Lage, den Index zu einem beliebigen Bereich einer relativen Datei (der natürlich alphabetisch geordnet sein muß) zu erstellen.

Das Programm ist anwenderfreundlich – es erkennt unlogische Eingaben und erklärt sich nach Programmstart selbst.

Es lohnt sich eigentlich gar nicht, den Index-File jedesmal erst lange von Disk zu laden. Es ist günstiger, die Werte einfach in DATA-Zeilen im Listing abzulegen.

Durch Anwendung der Index-Binären-Methode sind nur wenig Lesezugriffe auf Disk notwendig, bis ein Wort gefunden ist: Die größte Wortgruppe unserer Wortschatzdatei stellen die Wörter dar, die mit S beginnen: Das sind 33 Worte. Im ungünstigsten Fall werden nur 6 Zugriffe benötigt, um ein Wort zu finden. Vorteilhaft an der neuen Methode ist außerdem, daß Wörter, die mit einem Buchstaben beginnen, der im Wortschatz als Anfangsbuchstabe überhaupt nicht auftaucht (zum Beispiel C,J,M,X und Y), erst gar nicht gesucht werden, weil von vorneherein klar ist, daß sie im Wortschatz nicht auftreten.

Bei Methode 4 wird allerdings auch noch etwas anderes verbessert: Das Zerlegen von BE$ in BE$(0)–BE$(N). Eigentlich ist es nämlich Quatsch, BE$ in einzelne Worte zu zerlegen. Es reicht völlig aus, mit Zeigern zu arbeiten. Bild 9 verdeutlicht dies.

Das mit den Zeigern funktioniert ganz einfach: Zu Beginn wird Zeiger A auf das erste Zeichen von BE$ gesetzt. Zeiger B läuft von Zeiger A aus nach rechts bis zum nächsten Space. Das erste Suchwort ist somit festgelegt. Nachdem dieses Wort codiert ist, wird Zeiger A auf das Zeichen rechts neben Zeiger B gesetzt – auf das erste Zeichen des neuen Suchwortes. Jetzt läuft Zeiger B wieder nach rechts und sucht erneut das Wortende, und so weiter.

Trifft Zeiger B auf kein Zeichen mehr, so ist er am Ende des Satzes angelangt – der Satz ist komplett codiert.

Damit das letzte Wort der Sätze noch codiert wird, muß am Befehlssatz BE$ noch ein SPACE angehängt werden.

Bei Methode 4 soll auch noch etwas anderes berücksichtigt werden – das Satzende. Es gibt 3 Möglichkeiten, einen Satz zu beenden – mit einem Punkt, einem Ausrufezeichen, einem Fragezeichen oder einfach nichts.

Uns interessieren allerdings nur zwei Fälle: Fragezeichen oder nicht. Bevor BE$ also codiert wird, muß künftig darauf geachtet werden, ob am Ende des Satzes ein Satzzeichen steht. Die Variable SN speichert das Satzzeichen (Punkt etc. =0, Fragezeichen =1). Dann wird BE$ so gekürzt, daß das Zeichen wegfällt.

Nach dem Codiervorgang müssen folgende Werte in Form von Variablen vorliegen:

| | |
|---|---|
| SN | Satzzeichen am Satzende |
| VE | Verbzahl |
| O | Objektzahl für erstes Objekt |
| O2 | Objektzahl für zweites Objekt (falls vorhanden) |
| RI | Richtung |
| UD | UND-Speicher |
| AD | Adjektiv-Zahl |

```
        BE$=BE$+" "
BE$= | N I M M  D A S  S C H W E R T |
       ↑
       A

10 B = A+1:IF MID$(BE$,B,1)=" " THEN END
20 IF MID$(BE$,B,1)<>" "  THEN B=B+1:GOTO20

     | N I M M  D A S  S C H W E R T |
       ↑       ↑
       A       B

30 SU$=MID$(BE$,A,B-A)
JETZT WIRD SU$ CODIERT
DANN: A=B+1:GOTO10
```

**Bild 9. Befehlscodierung mit Methode 4**

```
10 OPEN 1,8,3,"WORTSCHATZ":OPEN 15,8,15:RE
   M WORTSCHATZ-DATEI OEFFNEN               <125>
20 DATA 1,3,0,28,31,47,51,60,68,0,70,76,0,
   82,90,93,100,101,116,149,155,158         <160>
30 DATA 178,0,0,189                         <193>
35 DATA 198 :REM BUCHSTABE NACH Z !?!       <235>
40 DIM IN(26):FOR I=0 TO 26:READ IN(I):NEX
   T I                                      <193>
50 GOSUB 50000                              <062>
60 PRINT"SN="SN:PRINT"VE="VE:PRINT"O1="O1:
   PRINT"O2="O2:PRINT"UD="UD:PRINT"RI="RI   <017>
70 PRINT"AD="AD:PRINT:GOTO 50               <111>
50000 REM ******************************    <227>
50001 REM *                            *    <009>
50002 REM *       WORT-PARSER 4.0      *    <169>
50003 REM *                            *    <011>
50004 REM *      <C> 1986 BEI          *    <086>
50005 REM *                            *    <013>
50006 REM *       MICHAEL NICKLES      *    <082>
50007 REM *                            *    <015>
50008 REM ******************************    <235>
50010 REM BEFEHLSSATZEINGABE -------------
      ------------------------------------
      --                                    <069>
50011 :                                     <202>
50012 IF UD>0 THEN 50500:REM UND            <171>
50015 SL=80: REM BEFEHLSSATZLAENGE          <144>
50020 PRINT"{DOWN}":BE$="":POKE 198,0:POKE
       211,0:POKE 214,22:SYS 58732:PRINT"{
      YELLOW}@";                            <156>
50030 GET X$:IF PEEK(203)=1 THEN 50120      <226>
50040 IF X$=""THEN 50030                    <171>
50050 IF LEN(BE$)=0 AND ASC(X$)=20 THEN 50
      030                                   <148>
50060 I=ASC(X$):IF I<32 OR I>133 AND I<159
       THEN IF I<>20 THEN 50030             <131>
50070 IF LEN(BE$)=SL AND I<>20 THEN 50030   <128>
50080 BE$=BE$+X$                            <031>
50090 PRINT CHR$(20);X$;"@";                <108>
50100 IF I=20 THEN BE$=LEFT$(BE$,LEN(BE$)-
      2):GOTO 50030                         <101>
50110 GOTO 50030                            <001>
50120 PRINT"{LEFT,SPACE}"                   <227>
50400 REM SATZZEICHEN SUCHEN -------------
      ------------------------------------
      --                                    <103>
50401 :                                     <082>
50415 SN=0                                  <104>
50420 IF RIGHT$(BE$,1)=" "THEN BE$=LEFT$(B
      E$,LEN(BE$)-1):GOTO 50420             <233>
50430 I$=RIGHT$(BE$,1)                      <180>
50440 IF I$="."OR I$="!"THEN SN=0:BE$=LEFT
      $(BE$,LEN(BE$)-1):GOTO 50420          <010>
50450 IF I$="?"THEN SN=1:BE$=LEFT$(BE$,LEN
      (BE$)-1):GOTO 50420                   <144>
50500 REM BE$-ZEIGER SETZEN --------------
      ------------------------------------
      --                                    <199>
50501 :                                     <184>
50502 IF UD=0 THEN VE=0:RI=0:O1=0:O2=0      <012>
50504 IF UD>0 THEN RI=0:GOTO 50530          <199>
50505 IF LEFT$(BE$,1)=" "THEN BE$=RIGHT$(B
      E$,LEN(BE$)-1):GOTO 50505             <132>
50510 BE$=BE$+" "                           <232>
50520 ZA=1 :REM ZEIGER A SETZEN             <058>
50530 ZB=ZA+1                               <180>
50540 IF MID$(BE$,ZB,1)=""THEN UD=0:RETURN  <213>
50550 IF MID$(BE$,ZB,1)<>" "THEN ZB=ZB+1:G
      OTO 50550                             <175>
50590 SU$=MID$(BE$,ZA,ZB-ZA)                <222>
50600 REM SU$ VORBEHANDLUNG --------------
      ------------------------------------
      --                                    <130>
50601 :                                     <028>
50610 IF SU$="UND"THEN UD=1:GOTO 50630      <093>
50620 IF SU$=","THEN UD=2:GOTO 50630        <061>
50625 GOTO 50670                            <201>
50630 ZA=ZB+1                               <080>
50640 IF MID$(BE$,ZA,1)=""THEN PRINT"UND W
      AS ?":GOTO 50530                      <153>
50650 IF MID$(BE$,ZA,1)=" "THEN ZA=ZA+1:GO
      TO 50650                              <242>
50660 RETURN                                <169>
50670 IF SU$="IHN"OR SU$="SIE"OR SU$="ES"T
      HEN O1=OM:GOTO 51300                  <069>
51000 REM BINAERE SUCHROUTINE ------------
      ------------------------------------
      --                                    <113>
51001 :                                     <176>
51007 REM ANFANG UND ENDE DES SUCHBEREICHE
      S ERMITTELN                           <117>
51008 I=ASC(LEFT$(SU$,1))                   <176>
51009 IF I-65<0 OR I-65>25 THEN GOSUB 5150
      0:GOTO 50000                          <134>
51010 AN=IN(I-65):REM ANFANG DES SUCHBEREI
      CHES                                  <164>
51012 IZ=64                                 <200>
51013 IF IN(I-IZ)=0 THEN IZ=IZ-1:GOTO 5101
      3                                     <086>
51015 EN=IN(I-IZ)-1:REM ENDE DES SUCHBEREI
      CHES                                  <070>
51016 IF AN=0 THEN GOSUB 51500:UD=0:GOTO 5
      0000                                  <249>
51020 SZ=197:N=INT(LOG(EN-AN+1)/LOG(2))+1
                     :REM MAXIMAL-FORMEL    <121>
51030 SA=AN-1+(2↑N)/2:REM MITTE DER GES.DA
      T                                     <073>
51040 GOSUB 52100:REM A$ LESEN              <234>
51050 N=N-1        :REM 1.ABFRAGE           <045>
51060 :                                     <235>
51070 REM SU$ MIT A$ VERGLEICHEN ------     <042>
51080 : IF LEN(SU$)<3 THEN 51110            <140>
51090 : IF WA=1 AND SU$=LEFT$(A$,LEN(SU$))
       THEN VE=WC:GOTO 51300                <157>
51100 : IF WA<>3 OR SU$<>RIGHT$(A$,LEN(SU$
      ))THEN 51110                          <161>
51102 :   IF UD=1 OR UD=2 THEN UD=3:O1=0:O2
      =0                                    <042>
51104 :    IF O1=0 THEN O1=WC:OM=O1:GOTO 51
      300                                   <133>
51105 :    IF O2=0 THEN O2=WC:OM=O1:GOTO 51
      300                                   <170>
51110 : IF WA=2 AND SU$=A$
       THEN RI=WC:GOTO 51300                <140>
51115 : IF WA=5 AND SU$=A$
       THEN AD=WC:GOTO 51300                <247>
51120 : IF WA=1 AND SU$=A$
       THEN VE=WC:GOTO 51300                <017>
51125 : IF WA=4 AND SU$=A$
       THEN 51300                           <204>
51130 : IF SU$<A$THEN SA=SA-(2↑(N-1)):GOSU
      B 52100:GOTO 51160                    <015>
51140 : IF SA+(2↑(N-1))>SA+EN-AN AND N>-1
      THEN N=N-1:GOTO 51140:REM ''UEBERLAU
      F''                                   <204>
51150 : SA=SA+(2↑(N-1)):GOSUB 52100:REM ''
      HALBIEREN''
                                            <141>
51160 :                                     <079>
51170 :N=N-1       :REM NAECHST. HALBIEREN  <068>
51180 :IF N<0 THEN GOSUB 51500:UD=0:GOTO 5
      0000                                  <201>
51190 REM PRINT"SA=";SA;" N=";N
                          :REM UNNOETIG !!  <216>
51200 :GOTO 51060:REM NAECHST. SUCH-VERS.   <192>
51300 REM NAECHSTES WORT                    <027>
51310 ZA=ZB+1                               <254>
51320 IF MID$(BE$,ZA,1)=""THEN 50530        <128>
51330 IF MID$(BE$,ZA,1)=" "THEN ZA=ZA+1:GO
      TO 51330                              <207>
51350 GOTO 50530                            <131>
51500 REM ICH KENNE SU$ NICHT ------------
      ------------------------------------
      --                                    <232>
51501 :                                     <168>
51510 I=INT(3*RND(1))+1                     <080>
51520 ON I GOTO 51530,51540,51550           <173>
51530 X$=""+"ICH KENNE DAS WORT '"+SU$+"'
      NICHT.":GOTO 52000                    <002>
51540 X$=""+"DAS WORT '"+SU$+"' IST MIR NI
      CHT BEKANNT.":GOTO 52000              <113>
51550 X$=""+"DEUTSCH IST LEIDER NUR MEINE
      ZWEITE SPRACHE. ICH KENNE DAS "       <038>
51555 X$=X$+"WORT '"+SU$+"' NICHT.":GOTO 5
      2000                                  <157>
52000 REM FORMATIERTE TEXTAUSGABE --------
      ------------------------------------
      --                                    <120>
52010 IF X$=""THEN 52090                    <213>
52020 IF LEN(X$)<40 THEN PRINT X$:GOTO 520
      90                                    <248>
52030 IF LEN(X$)=40 THEN PRINT X$;:GOTO 52
      090                                   <148>
52040 I$=LEFT$(X$,41):IC=0:FOR I=1 TO LEN(
      I$):IF MID$(I$,I,1)=" "THEN IC=I      <242>
```

```
52050 NEXT I:IF IC=0 THEN PRINT X$:GOTO 52
      090                                  <064>
52060 I$=LEFT$(I$,IC-1):IF LEN(I$)=40 THEN
       PRINT I$;                           <046>
52070 IF LEN(I$)<40 THEN PRINT I$          <116>
52080 X$=RIGHT$(X$,LEN(X$)-IC):GOTO 52010  <206>
52090 RETURN                               <075>
52100 REM SATZ-TEILBEREICH LESEN ----------
      -------------------------------------
      --                                   <254>
52101 :                                    <004>
52110 HB%=SA/256:LB%=SA-HB%*256            <241>
52112 PRINT#15,"P";CHR$(3);CHR$(LB%);CHR$(
      HB%);CHR$(1)                         <170>
52114 INPUT#15,A:IF A=50 THEN PRINT"SATZNU
      MMER ZU HOCH"                        <245>
52120 IF A=50 THEN RETURN                  <165>
52130 INPUT#1,A$                           <199>
52140 WA=VAL(MID$(A$,22,1))                <030>
52150 WC=VAL(MID$(A$,23,3))                <089>
52160 A$=LEFT$(A$,21)                      <202>
52170 IF RIGHT$(A$,1)="{RIGHT}"THEN A$=LEF
      T$(A$,LEN(A$)-1):GOTO 52170          <154>
52180 RETURN                               <165>
```

**Listing 8. Methode 4 als Programm**

Zur Objektcodierung werden zwei Variablen O1 und O2 benötigt, da ein Befehl aus maximal zwei Objekten (direktes und indirektes) bestehen kann. Beispiel: »ÖFFNE DIE TÜR MIT DEM SCHLÜSSEL«. Im Befehl kann auch nur ein Verb auftreten. Gegebenenfalls kann auch noch eine Variable AD für Adjektiv zur näheren Beschreibung der Objekte verwendet werden. Adjektive haben die Wortzahl WZ=5.

Jetzt kann Methode 4 wie in Listing 8 programmiert werden:

Die REM-Zeilen können beim Abtippen nicht so ohne weiteres weggelassen werden!

Mit REM-Zeilen beansprucht Parser 4.0 zirka 3 KByte RAM Basic-Speicher. Entfernt man die REM-Zeilen unter Berücksichtigung der dann teils veränderten Ansprungsadressen, so werden nur zirka 2 KByte Basic-RAM benötigt.

Parser 4.0 ist ziemlich intelligent und zum größten Teil sogar idiotensicher. Das heißt, es werden die blödsinnigsten Befehlseingaben akzeptiert. Zum Beispiel:

```
"NIMM DAS SCHWERT"
" NIMM  DAS  SCHWERT!"
"NIMM SCHWERT, UNTERSUCHE ES"
```

Die UND-Routine ist auch eingebaut. Außerdem speichert Parser 4.0 jeweils das zuletzt eingegebene Objekt in der Variablen OM. Bei Eingabe eines der Schlüsselwörter IHN, SIE oder ES (zum Beispiel BE$=»UNTERSUCHE ES«), so wird O gleich OM gesetzt. Der Clou ist auch, daß Sie einfach nur GOSUB 50000 eingeben müssen, wenn vom Spieler eine Befehlseingabe erwartet wird. Parser 4.0 erkennt automatisch, ob die INPUT-Routine angesprungen werden muß oder ob der letzte Befehlssatz noch nicht ganz abgehandelt ist (UND-FALL). Kann Parser 4.0 ein Wort nicht codieren, weil es im Wortschatz nicht vorkommt, so erfolgt »ICH KENNE DAS WORT .... NICHT«, und anschließend wird zur Befehlseingabe-Routine zurückgesprungen. Dies ist notwendig, damit ein Spieler niemals verlieren kann, wenn er bei einer entscheidenden Situation (zum Beispiel Kampf mit Monster) sich aus Versehen vertippt (zum Beispiel TOTE MONSTER statt TOETE MONSTER). Das Programm darf also erst dann weiterlaufen, wenn der Spieler eine korrekte Befehlseingabe gemacht hat.

Nach dem Rücksprung aus Parser 4.0 liegt der Satz codiert in den Variablen SN,VE,O1,O2,UD,AD vor. Natürlich können Sie Parser 4.0 jetzt noch nicht gründlich testen, da im Wortschatz momentan noch die Objekte fehlen. Zunächst rate ich Ihnen, die nun folgende Dokumentation gründlich zu studieren, damit Sie in der Lage sind, Parser 4.0 selbst zu modifizieren – denn dies ist beim Einbau in eigene Programme fast immer notwendig!

## Dokumentation zu Parser 4.0

10 Hier wird die Wortschatz-Datei geöffnet. Das heißt es muß die Diskette eingelegt sein, auf der sich der Wortschatz befindet.

20–30 Hier stehen die DATAs zum Wortschatz-Index, die mit dem Index-Maker ermittelt wurden.

35 Tatsächlich besteht die Index-Tabelle nicht nur aus 26, sondern 27 Elementen. Das 27. Element steht für den Buchstaben, der nach Z kommt. Richtig, nach Z kommt kein Buchstabe mehr. Das 27. Element der Index-Tabelle errechnet sich vielmehr aus ANZAHL WÖRTER DER WORTTABELLE +1. Dies ist nötig, damit das Programm erkennt, von welchem bis zu welchem Satz die mit Z beginnenden Wörter zu finden sind. Der 27. Wert wird NICHT automatisch vom Index-Maker erstellt. Er muß als »von Hand« berechnet und eingesetzt werden.

40 Einlesen der Index-Datas in das Index-Feld IN(0)–IN(26)

50–70 Hier wird Parser 4.0 mittels GOSUB 50000 aufgerufen. Danach werden die codierten Werte, SN, VE, 01, 02, UD, RI, AD ausgegeben. Dann erfolgt Rücksprung zum Parser-Aufruf. Die Zeilen 50–70 dienen also nur zum Testen des Parsers.

50000–50120 Hier erfolgt die Befehlseingabe in Form einer erweiterten INPUT-Routine, wie ich sie Ihnen im Kursabschnitt Datei-Verwaltung bereits vorgestellt habe.

50012 Hier wird überprüft, ob die letzte Befehlseingabe bereits voll codiert worden ist oder ob sie nach einem UND abgebrochen wurde. ZB BE$=»NIMM SCHWERT UND GEH NORDEN« wobei erst der Teilbefehl »NIMM SCHWERT« ausgeführt worden ist. Ist der Befehlssatz BE$ also noch nicht komplett codiert, so wird die Befehlseingabe (und natürlich auch die Suche nach dem Satzzeichen, die ja bereits erfolgt ist) übersprungen, alle bisherigen codierten Werte VE,01 etc. werden beibehalten, und es wird dort weiter codiert, wo zuletzt abgebrochen wurde - GOTO 50500.

50400-50450 Hier wird nach einem Satzzeichen am Ende von BE$ gesucht.

50420 Alle unnötigen Leerzeichen, die dem Befehlssatz BE$ anhängen, werden weggekürzt. Beispiel: AUS BE$=»NIMM SCHWERT ! « wird BE$=»NIMM SCHWERT!«.

50440–50450 Wird ein Satzzeichen gefunden, so wird es in der Variablen SN gespeichert (interessant ist dabei eher nur das Fragezeichen!). Dann wird das Satzzeichen weggekürzt. Also aus »NIMM SCHWERT !« wird jetzt »NIMM SCHWERT«. All dies ist notwendig, damit bei einer Befehlseingabe BE$=»NIMM SCHWERT!« nicht die Fehlermeldung » »Ich kenne das Wort SCHWERT!« nicht« nicht auftritt. Denken Sie also stets daran, alle unnötigen Leerzeichen vor oder nach dem Befehl zu eliminieren. Einen Parser zu programmieren, ist wirklich eine verflixte Sache: Es gibt soviele Dinge, die berücksichtigt werden müssen, damit der Parser auch wirklich absturzfrei läuft. Das Wegkürzen der Leerstellen ist ein gutes Beispiel hierfür. Ein Tip: Wenn Sie glauben, den Parser endlich fertig zu haben, dann compilieren Sie ihn mit Hilfe eines Compilers wie zum Beispiel Austro Comp oder Pet Speed. Der Compiler überprüft alle Zeichen und weist auf eventuelle Syntax-Errors hin. Wird ein Parser-Programm fehlerfrei compiliert, so können Sie davon ausgehen, daß im Programm keine schwerwiegenden Fehler vorkommen!

50500–50600 Hier werden die Zeiger ZA und ZB auf den Befehlssatz BE$ gesetzt oder weitergerückt.

50502 Wurde beim letzten Vorgang nicht wegen UND abgebrochen, so werden alle bisher codierten Werte auf 0 gesetzt.

50504 Wird hier festgestellt, daß der letzte Befehlssatz noch nicht komplett codiert worden ist (UD größer 0), so passiert folgendes: Die Richtungszahl RI wird auf 0 gesetzt (warum, erfahren Sie später noch!). Dann erfolgt ein Sprung zu Zeile 50530. Das heißt, alle Operationen oder Einstellungen, die vorgenommen werden, wenn ein neuer Befehlssatz codiert werden soll (also kein wegen UND abgebrochener), werden übersprungen.

50505–50520 Einstellung der Zeiger auf einen neuen Befehlssatz.

50505 Alle Leerstellen, die dem eigentlichen Befehl voranstehen, werden entfernt. Beispiel: Aus »     NIMM SCHWERT« wird »NIMM SCHWERT«.

50510 BE$ wird ein Leerzeichen angehängt. Aus BE$ = »NIMM SCHWERT« wird also BE$ = »NIMM SCHWERT «. Dies ist nötig, da später beim Suchvorgang das Auffinden eines Leerzeichens als Markierung eines Wortendes interpretiert wird.

50520 Zeiger ZA wird auf das erste Zeichen von BE$ gesetzt.

50530 Zeiger ZB wird auf das Zeichen nach ZA gesetzt.

50540 Ist MID$(BE$,ZB,1) = "", so wurde das Satzende erreicht. Der Codiervorgang wird mittels RETURN abgebrochen.

50550 Ist MID$(BE$,ZB,1) kein Leerzeichen, so wird Zeiger ZB um ein Zeichen nach rechts gerückt und es erfolgt ein Rücksprung zu Zeile 50550. Zeiger ZB wird also so lange nach rechts geschoben, bis entweder das Satzende oder ein Leerzeichen gefunden wird. Wird ein Leerzeichen gefunden, so ist das neue Suchwort festgelegt: SU$=MID$(BE$,ZA,ZB–ZA).

50590 Das neue Suchwort, das von den Zeigern ZA und ZB auf BE$ bestimmt ist, wird im String SU$ abgelegt.

50600–50670 Es erfolgt eine »Vorbehandlung« des Suchwortes SU$. Hierbei werden solche Wörter gesucht, die Parser-intern gespeichert sind und nicht auf der Wortschatz-relativen Datei verzeichnet sind.

50610 Hier wird überprüft, ob SU$ gleich UND ist. Wenn ja, so ist klar, daß ein Teilbefehl von BE$ codiert ist und später von hier aus weitercodiert werden muß. Zunächst muß jedoch der bisherig ermittelte Teilbefehl analysiert werden. Dazu wird die Variable UD auf 1 gesetzt und der Codiervorgang mittels RETURN abgebrochen.

50620 Das gleiche wie bei 50610 in Grün – nur, daß hier nach einem Komma gesucht wird. Die Komma-Suche hat allerdings einen Haken: Das Komma wird nur dann gefunden, wenn links und rechts von ihm mindestens je ein Leerzeichen steht. Bei Befehlssätzen wie zum Beispiel BE$ = »NIMM SCHWERT,BIERFLASCHE« erfolgt eine Fehlermeldung à la »Ich kenne das Wort Bierflasche nicht«. Mit einem direkten Vergleich IF SU$ = "," ist es also nicht abgetan. Um den Parser nicht unnötig kompliziert zu machen, habe ich deshalb auf diese umfangreiche Abfrage verzichtet. Wie umfangreich diese Abfrage ist, werden Sie spätestens dann bemerken, wenn Sie versuchen, sie zu programmieren. Ich rate Ihnen, bei einem Basic-Parser auf diese Abfragerei zu verzichten, denn sie kostet viel zuviel kostbare Rechenzeit, und der Parser wird zu langsam!

50630–50660 Diese Zeilen treten dann in Kraft, wenn ein UND oder ein Komma gefunden worden ist. Zeiger ZA wird auf eine Stelle nach ZB gesetzt.

50540–50550 Hier erfolgt das bereits beschriebene Weiterrücken von Zeiger ZB nach rechts, bis ein Leerzeichen oder das Satzende gefunden wird. Wird hier ein Satzende festgestellt, so heißt dies, daß nach dem UND oder Komma nichts mehr kommt. Der Parser äußert sich im Fall zum Beispiel BE$ = »NIMM SCHWERT UND« empört mit »Und was?«, wobei er wieder einen Hauch von Intelligenz beweist und den Spieler in Erstaunen versetzt ...

50670 Wird eines der Schlüsselwörter IHN, SIE oder ES gefunden, so weiß der Parser sofort, daß der Spieler sich auf das zuletzt erwähnte Objekt bezieht. Das zuletzt erwähnte Objekt wird immer in der Variablen OM gespeichert. Der Parser braucht hier also nichts weiter zu tun, als der Objektzahl 01 den Wert OM zuzuweisen und dann die binäre Suchroutine zu überspringen mit GOTO 51300.

51000–51330 Hier steht die binäre Suchmethode, mit deren Hilfe SU$ in der relativen Wortschatzdatei gesucht wird.

51008 Der ASC-Wert des 1. Buchstabens vom Suchwort SU$ wird in der Variablen I gespeichert.

51009 Wenn I nicht der ASC-Wert eines der Buchstaben von A–Z ist, also I–65 kleiner 0 oder I–65 größer 25, dann ist klar, daß das Suchwort nicht mit einem Buchstaben, sondern mit irgendeinem Zeichen beginnt, zum Beispiel SU$="/%&%/". Es erfolgt dann eine »ICH KENNE DIESES WORT NICHT«-Meldung, und der Codiervorgang wird abgebrochen.

51010 Der Anfang des zu durchsuchenden Bereiches wird mittels der Index-Tabelle IN(0)–IN(26) ermittelt und in AN gespeichert.

51012–51015 Hier wird das Ende des Suchbereiches, EN ermittelt. Das Ende berechnet sich NICHT einfach durch EN=IN(I–64), da berücksichtigt werden muß, daß in der Index-Tabelle für nicht existierende Anfangsbuchstaben (zum Beispiel C) eine Null eingetragen ist. Es darf also nicht der unmittelbare nächste Buchstabe zur Ermittlung des Suchendes herangezogen werden, sondern vielmehr der nächste existente Buchstabe (Tabellenwert ungleich 0).

51016 Wenn AN=0 ist, so beginnt das Suchwort SU$ mit einem Buchstaben, mit dem kein Wort des Wortschatzes beginnt. Es braucht dann gar nicht erst lange gesucht werden – Fehlermeldung ICH KENNE WORT NICHT.

51020–51200 Hier steht die binäre Suchroutine, die ich Ihnen bereits im Kapitel Dateiverwaltung beschrieben habe. Allerdings ist die Routine diesmal so modifiziert, daß sie mit relativen Dateien arbeitet und nicht mit Stringfeldern im Speicher.

52100–52180 Dieses Unterprogramm wird innerhalb der binären Suchroutine angesprungen. Es dient dazu, um aus einer relativen Datei einen Satz zu lesen. Diese Routine ist im wesentlichen eine Kombination unserer bisher verwendeten Unterprogramme zum Positionieren auf (GOSUB 9000) und Lesen (GOSUB 9100) eines Satzes aus einer relativen Datei.

52100–52130 Positionieren und anschließendes Lesen des Satzes A$.

52140 Die Wortart WA wird ermittelt.

52150 Der Wortcode WC wird ermittelt.

52160 AS wird auf das eigentliche Wort gekürzt. Aus A$=»abwaerts          2010« wird A$=»abwaerts          «.

52170 Alle unnötigen Leerstellen in A$ werden gekürzt. Aus A$=»abwaerts          « wird also A$=»abwaerts«. Dies ist eigentlich nur dann nötig, wenn es sich bei A$

um ein Objekt handelt (WA=3), denn schließlich werden nur Objekte »rechtsbündig« verglichen beim Suchvorgang. Wer will, kann folgende Zeile einfügen (ohne Gewähr):

```
52165 IF WA=3 THEN RETURN
```

Warum ohne Gewähr? Während man einen Parser programmiert oder dokumentiert, entdeckt man ständig kleine Dinge, die man noch zu verbessern glaubt. Schnell sind drei, vier Änderungen gemacht, und plötzlich läuft der Parser nicht mehr so recht, obwohl alle Änderungen für sich allein betrachtet doch eigentlich funktionieren sollten!?! Ich rage Ihnen daher, an einem Parser-Programm nie mehr als eine Änderung auf einmal vorzunehmen und dann auch gleich alle Fälle auszuprobieren, um sicherzustellen, daß auch alles noch funktioniert...

51070-51125 Hier werden die einzelnen Wortzahlen VE,01 etc. ermittelt. Es wird also genau hier codiert.

51080 Verben werden, wie bereits erklärt, linksbündig gesucht beim Vergleich - also IF SU$=LEFT$(A$, LEN(SU$))...
Es sind somit auch Abkürzungen legitim - zum Beispiel NIM statt NIMM oder INV statt INVENTUR. Eine Abkürzung muß allerdings mindestens 3 Zeichen lang sein. Ist die Abkürzung kleiner als 3 Zeichen, so werden Verbsuche und Objektsuche übersprungen und der Parser stellt sich darauf ein, eine Richtungsangabe zu finden. Würde man auf diese Abfrage verzichten, so wüßte der Parser nicht, wie er sich im Fall einer Befehlseingabe BE$=»N« verhalten soll. Er würde als Verbzahl VE das nächstbeste Verb ermitteln, das mit N beginnt, und anschließend auch noch die Richtungsangabe beziehungsweise Richtungszahl RI für die Richtung Norden ermitteln.

51090 Hier erfolgt die linksbündige Verbsuche.

51100 Hier erfolgt die Objektsuche - und zwar wie bereits erklärt rechtsbündig, damit zum Beispiel sowohl HOLZKISTE als auch KISTE verstanden werden, wenn der Objektname HOLZKISTE lautet.

51102 Jetzt wird's kompliziert zu erklären!
Es wird überprüft, ob bei der letzten Codierung wegen Auffinden eines UNDs beziehungsweise Kommas abgebrochen wurde (UD größer 0). Ist dies der Fall, so werden 01 und 02 gelöscht und UD auf 3 gesetzt, damit beim Auffinden eines zweiten Objekts nicht wieder gelöscht wird.

51104-51105 In einem Befehlssatz können maximal zwei Objekte gleichzeitig auftreten (beziehungsweise Teilbefehl) - vergleiche Wortschatzlehre! Es müssen also zwei Variablen zum Speichern von Objekten zur Verfügung stehen - 01 und 02.
Nehmen wir einmal BE$=»ÖFFNE TÜR MIT DEM SCHLÜSSEL UND BETRETE DAS ZIMMER« an.
Zuerst wird ÖFFNE im Wortschatz gesucht und eine Verbzahl VE gefunden. Dann wird das Wort TÜR gesucht und gefunden. Das Objekt TÜR wird codiert in der 1. Objektzahl 01 gespeichert.
Das Wort MIT wird nicht berücksichtigt (ich setze hier einmal voraus, daß es im Wortschatz als AUS-Wort aufgeführt ist).
Jetzt wird das Objekt SCHLÜSSEL gesucht und gefunden. In der Variablen 01 kann sein Code nicht gespeichert werden, da 01 bereits belegt ist. In 02 ist allerdings noch Platz. Dem Parser ist also klar, daß ein zweites Objekt gefunden wurde und dies in 02 gespeichert werden muß. Nehmen wir jetzt einmal den unmöglichen Fall an, daß in einem Befehlssatz oder Teilbefehl 3 Objekte auftauchen. Das dritte Objekt wird vom Parser dann eiskalt ignoriert, und es erfolgt keine Fehlermeldung. Wenn in einem Befehlssatz 3 Objekte verwendet werden, so ist klar, daß der Befehlssatz nur unsinnig sein kann, zum Beispiel »ERSCHLAGE MONSTER MIT DEM SCHWERT DAS AUF DEM STUHL LIEGT«. Ok, eigentlich ergibt dieser Satz einen Sinn, aber wir wissen, daß dies nicht die Eingabe-Syntax eines routinierten Adventure-Spielers ist. Man dürfte es dem Parser also nicht übel nehmen, wenn er sich mit »ICH VERSTEHE SIE NICHT« meldet. Man könnte also (wieder ohne Gewähr) noch folgende Zeile einbauen:

```
51103 IF 01<>0 AND 02<>0 THEN? »ICH
VERSTEHE SIE NICHT«:GOTO50000.
```

Als nächstes Wort wird UND gefunden. Die Variable UD wird auf 1 gesetzt und der Codiervorgang abgebrochen. Der erste Teilbefehl des gesamten Befehlssatzes ist somit codiert und liegt anhand der Variablen VE,01,02 und sekundär UD vor.
Nach Analyse dieses Teilbefehls wird die Codierung fortgesetzt. Zunächst wird BETRETE als neues VERB ermittelt. Aus der Wortschatzlehre wissen Sie bereits, daß ein Befehlssatz maximal ein Verb enthalten kann. Stehen unsinnigerweise doch mal zwei Verben in einem Befehl, so speichert Parser 4.0 den Wert des Verbs in VE ab, das am weitesten rechts im Befehl steht.
Jetzt wird nach Ignorieren von DAS das neue Objekt ZIMMER gesucht. Wäre Zeile 51102 nicht vorhanden, so würde der Parser dieses Objekt nicht mehr codieren können, da erst die beiden Werte 01 und 02 gelöscht werden müssen.
Jetzt liegt der zweite Teilbefehl von BE$ codiert vor. Und zwar diesmal in den Variablen VE und 01.

51110 Hier wird eine Richtungsangabe absolut gesucht.

51115 Adjektivsuche - ebenfalls absolut.
Zur Adjektivcodierung wurde nur die Variable AD reserviert. Ein Befehl kann also nur maximal ein Adjektiv enthalten. Dies langt in der Regel. Wer jedoch Adventures programmieren will, bei denen extrem komplizierte Befehlseingaben als BE$=»ÖFFNE DIE GRÜNE TÜR MIT DEM GELBEN SCHLÜSSEL« erforderlich sind, der muß hier entsprechend modifizieren (analog zur Objektcodierung mittels 01 und 02).

51120 Absolute Verbsuche. Hier werden Verben gefunden, deren Satzlänge weniger als 3 Zeichen beträgt. Wollen Sie zum Beispiel I als Abkürzung für INVENTUR erlauben, so können Sie I extra in der Wortschatzdatei als Verb eintragen.

51125 AUS-Wörter wie der, die, das werden absolut gesucht und gegebenenfalls ignoriert.

51300-51350 Weiterrücken der Zeiger ZA und ZB aufs nächste Wort. Eine neue Suchrunde wird eingeläutet.

51500-51555 Kleine Bierbraukunde

52000-52090 Wird hier endlich geklärt, was Lolo, Peter und Mike im Wald machen?

**Variablenliste**

IN(0) bis IN(26) Index-Tabelle

| | |
|---|---|
| UD | UND-Variable (größer 0 bei Codierungsabbruch wegen UND oder ,) |
| SL | Maximal erlaubte Befehlssatzlänge (max. 255 Zeichen!) |
| BE$ | Gesamter Befehlssatz |
| X$ | Hilfsstring |
| I | Hilfsvariable |
| SN | Satzzeichenwert |
| I$ | Hilfsstring |
| VE | Verbzahl |
| RI | Richtungszahl |
| 01 | 1. Objektzahl |

| | |
|---|---|
| O2 | 2. Objektzahl |
| ZA,ZB | Zeiger zum Eingrenzen des Suchwortes SU$ in BE$ |
| SU$ | Suchwort |
| IZ | Hilfsvariable |
| AN | Anfang des Suchbereichs |
| EN | Ende " " |
| SZ | Anzahl der Worte in der Wortschatzdatei |
| N | Hilfsvariable |
| SA | Satz einer relativen Datei (Nummer) |
| WA | Wortart |
| WC | Wortcode |
| HB% | Higher Byte |
| LB% | Lower Byte |
| A | Fehlerkanal-Zahl |
| A$ | Satz, der von relativer Datei gelesen wurde. |

Das war's auch schon. Anhand dieser umfangreichen Dokumentation sollte es Ihnen nun nicht mehr schwerfallen, sich mit Parser 4.0 zurechtzufinden und eigene Änderungen am Programm durchzuführen. Mit Sicherheit haben Sie bald so manche Idee, was man noch verbessern kann. Ich rate Ihnen auf jeden Fall, an dieser Stelle erst dann mit dem folgenden Kursabschnitt fortzufahren, wennn Sie Parser 4.0 verstanden haben.

## Für Profis: Parser 4.1

Parser 4.1 ist eine Verbesserung beziehungsweise eine schnellere Version von Parser 4.0. Das nun Folgende soll Ihnen lediglich als theoretisches Gedankengut dienen, wenn Sie Parser 4.0 selbständig verbessern wollen.

Parser 4.0 basiert auf dem index-binären Prinzip. Der Wortschatz muß daher alphabetisch geordnet sein, damit eine Index-Tabelle IN(0)–IN(26) erstellt werden kann. Aus dieser Tabelle kann man schnell erlesen, in welchem Satzbereich der relativen Datei Wörter mit gleichem Anfangsbuchstaben stehen. Durch diese Eingrenzung des Suchbereiches werden einzelne Suchwörter mit relativ wenigen Diskzugriffen gefunden. Die Idee von Parser 4.1 ist wie folgt: Auch hier findet die index-binäre Methode Verwendung. Allerdings wird diesmal nicht eine Indextabelle IN(0)–IN(26) verwendet, sondern vielmehr ein Index-Feld, das so angelegt wird:

```
DIM IN%(26,26)
```

In IN%(0,0)–IN%(26,0) steht hierbei wieder die gleiche Index-Tabelle, die bei Parser 4.0 in IN(0)–IN(26) steht.

Jetzt spielt allerdings auch noch der 2. Buchstabe des Suchwortes eine Rolle bei der Einschränkung des Suchbereiches.

In IN%(0,0)–IN%(0,26) zum Beispiel ist die Index-Tabelle des zweiten Buchstabens aller Wörter, die mit A beginnen, gespeichert.

Das heißt, in IN%(0,0) steht, wo alle Wörter der Wortschatzdatei beginnen, deren erste beide Buchstaben AA sind. In IN%(0,1) entsprechend AB, in IN%(1,3) entsprechend BD etc. Anhand eines Index-Feldes kann der Suchbereich folglich noch viel feiner eingeschränkt werden als bei einer Index-Tabelle. Das heißt, es werden weniger zeitraubende Disketten-Zugriffe zum Finden eines Suchwortes benötigt, und der gesamte Parser wird schneller.

Parser 4.0 braucht ziemlich lange, um festzustellen, daß er ein mit den beiden Buchstaben RT beginnendes Wort nicht kennt. Parser 4.1 hingegen benötigt nicht einmal einen einzigen Disketten-Zugriff. Er kann aus seinem Index-Feld auf Anhieb erlesen, daß die Buchstabenkombination RT im Wortschatz nicht vertreten ist. Parser 4.0 läßt sich schnell und relativ einfach zu Parser 4.1 aufrüsten. Zunächst müssen Sie den Index-Maker modifizieren, damit er Index-Felder erstellen kann. Der Rest dürfte für Sie dann ein Kinderspiel sein – wenn Sie Parser 4.0 verstanden haben.

## Parser 5.0 – der Superparser

Parser 4.0 ist ja recht nett. Man kann relativ umkompliziert den Wortschatz erweitern und schnell Programmänderungen vornehmen. Trotzdem reicht Parser 4.0 in puncto Geschwindigkeit keineswegs an den INFOCOM-Parser heran. Parser 4.0 braucht im Durchschnitt 15 Sekunden zum Codieren eines Befehls, Parser 4.1 etwa die Hälfte. Mit dem Codieren ist es allerdings noch lange nicht abgetan – bis auf dem Bildschirm eine Antwort erscheint, muß die Codierung erst noch analysiert werden etc. Um eine INFOCOM ähnliche Geschwindigkeit zu erreichen, muß man daher einen möglichst schnellen Parser haben, um Zeit für die Analysenberechnung zu sparen. Wie macht's INFOCOM? Mit Sicherheit verwaltet INFOCOM seine 1000 Wörter NICHT in Form einer relativen Datei. Zwar beginnt nach jeder Befehlseingabe die Floppy zu laufen, jedoch nur, um den Antwortsatz einzulesen. Codiert ist er ruck, zuck. Um solche Geschwindigkeiten zu erreichen, hilf die schnellste Floppy beziehungsweise relative Datei-Programmierung nicht weiter. Der Wortschatz muß irgendwo im RAM untergebracht werden. Nehmen wir einmal an, wir möchten im Speicher 600 Wörter unterbringen. Für jedes Wort haben wir uns 21 Zeichen +4 Zeichen für Wortart und Wortcode reserviert. Wir bräuchten also 600 mal 25 Byte RAM, um diesen Wortschatz im RAM zu verstauen. Also einen freien RAM-Bereich von zirka 15 KByte. Abgesehen vom Basic-RAM verfügt der C64 über freies

```
10 REM ******************************       <022>
20 REM *         RAM-EDITOR         *       <053>
30 REM *       <C> 1985 BEI         *       <149>
40 REM *     MICHAEL NICKLES        *       <157>
50 REM ******************************       <062>
60 DIM M$(255),WZ(26)                       <152>
70 GOTO 51000                               <082>
8000 REM ORDNUNGSFILE INDEX-SEQ ----------
     ------------------------------------
     -                                      <087>
8005 PRINT"MOECHTEN SIE EINEN INDEX-FILE E
     RSTELLEN?";                            <144>
8006 GET I$:IF I$<>"J"AND I$<>"N"THEN 8006 <051>
8008 IF I$="N"THEN RETURN                   <240>
8019 INPUT"AB SPEICHERSTELLE ";E1           <124>
8020 INPUT"BIS{15SPACE}";E2:IF E1>E2 THEN
     PRINT"UNSINN!:GOTO8005                 <234>
8030 INPUT"SCHRITTWEITE-STEP ";SW           <048>
8100 PRINT"{DOWN}BITTE WARTEN!"             <255>
8105 FOR I1=E1 TO E2 STEP SW                <235>
8110 :FOR I2=1 TO 26                        <222>
8112 : IF USR(I1)=64+I2 AND WZ(I2)=0 THEN
     WZ(I2)=I1                              <001>
8114 :NEXT I2                               <198>
8120 NEXT I1                                <093>
8400 REM WZ(N) ALS SEQ-FILE SPEICHERN ----
     ------------------------------------
     -                                      <010>
8410 POKE 198,0:INPUT"FILENAME ";I$:I$=I$+
     ",S,W"                                 <231>
8420 OPEN 2,8,2,I$                          <135>
8430 FOR I=1 TO 26                          <249>
8440 :PRINT#2,WZ(I)                         <080>
8450 NEXT I:CLOSE 2                         <219>
8500 RETURN                                 <176>
9000 REM UP-POSITIONIEREN ----------------
     ------------------------------------
     -                                      <215>
9010 HB%=SA/256:LB%=SA-HB%*256              <068>
9020 PRINT#15,"P";CHR$(3);CHR$(LB%);CHR$(H
     B%);CHR$(1)                            <005>
9030 INPUT#15,A:IF A=50 THEN PRINT"SATZNUM
     MER ZU HOCH!":END                      <135>
9040 RETURN                                 <208>
9100 REM UP-SATZ LESEN -------------------
     ------------------------------------
     -                                      <037>
9110 IF A=50 THEN RETURN                    <082>
```

**Listing 9. Relative Dateien werden mit dem RAM-Editor in RAM-Dateien gewandelt**

```
9120 INPUT#1,A$                               <116>
9130 RETURN                                   <042>
51000 REM RAM-EDITOR ---------------------
      ------------------------------------
      --                                      <113>
51029 POKE 53280,0:POKE 53281,0:PRINT"{WHI
      TE}"                                    <100>
51030 PRINT"{CLR,RVSON}RAM-EDITOR V. 1.5{4
      SPACE}VON MICHAEL NICKLES{RVOFF}";      <049>
51040 REM MODIFIZIERTE PEEK-FUNKTION -----
      ------------------------------------
      --                                      <234>
51045 DATA 32,247,183,120,169,52,133,1        <112>
51048 DATA 160,0,177,20,168,169,55,133        <243>
51050 DATA 1,88,76,162,179                    <078>
51060 FOR I=704 TO 724:READ X:POKE I,X:S=S
      +X:NEXT I                               <178>
51070 POKE 785,192:POKE 786,2:REM USR-ZEIG
      ER                                      <022>
51080 IF S<>2325 THEN PRINT"FEHLER IN DATA
      S{SHIFT-SPACE}!":END                    <036>
51130 REM ANFANG -------------------------
      ------------------------------------
      --                                      <174>
51131 REM GOTO8000                            <127>
51132 PRINT"{DOWN}RAM EDITOR{2SPACE}KANN I
      N FREIEN RAM-BEREICHEN";                <128>
51134 PRINT"RELATIVE DATEIEN VERWALTEN."      <066>
51135 PRINT"BITTE NAME DER RELATIVEN DATEI
       ANGEBEN,                               <145>
51136 PRINT"VON DER DATEN INS RAM GESCHRIE
      BEN WERDEN";                            <246>
51137 PRINT"SOLLEN (DISK EINLEGEN)!"          <027>
51138 INPUT"{DOWN}FILENNAME ";F$              <028>
51139 OPEN 15,8,15  :REM FLOPPYKANAL AN       <130>
51140 PRINT#15,"I0":REM INITIALIZIEREN        <098>
51141 OPEN 1,8,3,F$ :REM DATEI OEFFNEN        <200>
51143 PRINT"BITTE GEBEN SIE ANFANG UND END
      E DES{5SPACE}BEREICHES AN, DER INS R
      AM";                                    <041>
51144 PRINT" GESCHRIEBEN{3SPACE}WERDEN SOL
      L!"                                     <037>
51146 POKE 198,0:INPUT"{DOWN}ANFANG";AN       <122>
51147 INPUT"ENDE{2SPACE}";EN:IF EN<=AN THE
      N PRINT"UNSINN !":GOTO 51146            <235>
51148 SA=AN:GOSUB 9000:GOSUB 9100             <143>
51149 PRINT"{DOWN}BITTE GEBEN SIE ANFANG U
      ND ENDE DES{5SPACE}RAM-BEREICHES AN,
       IN DEM";                               <088>
51150 PRINT" DIE DATEN{6SPACE}ABGELEGT WER
      DEN SOLLEN! ACHTUNG - DIESE ANGABEN
      ";                                      <041>
51151 PRINT"WERDEN NICHT UEBERPRUEFT !        <048>
51152 POKE 198,0:INPUT"{DOWN}ANFANG";E1       <029>
51153 INPUT"ENDE";E2                          <149>
51154 IF E2<E1 THEN PRINT"UNSINN !":GOTO 5
      1152 RUN                                <220>
51159 GOSUB 60000                             <122>
51200 REM INS RAM SCHREIBEN ----------        <023>
51205 PRINT"{DOWN}BITTE WARTEN!"              <179>
51210 IZ=E1                                   <177>
51215 FOR SA=AN TO EN:GOSUB 9000:GOSUB 910
      0:                    REM LESE A$       <188>
51220 FOR II=1 TO 3                           <176>
51225 :IF MA(II)=0 THEN 51250                 <152>
51230 :IF TY%(II)=2 THEN GOSUB 51500:REM $    <097>
51240 :IF TY%(II)=1 THEN GOSUB 51600:REM V
      AL                                      <213>
51250 NEXT II                                 <171>
51255 NEXT SA                                 <128>
51270 REM SPEICHERN AUF DISK ---------        <206>
51280 PRINT"{DOWN}BITTE DISKETTE EINLEGEN
      AUF DER RAM-                            <133>
51281 PRINT"BEREICH GESPEICHERT WIRD! [RET
      URN]":POKE 198,0:WAIT 198,1             <088>
51290 POKE 198,0:INPUT"FILENAME";I$:IF I$=
      ""THEN 51290                            <075>
51300 PRINT"{DOWN}BITTE WARTEN!"              <018>
51315 FOR I=1 TO 15:CLOSE I:NEXT              <206>
51320 OPEN 1,8,1,I$                           <202>
51330 HB%=INT(E1/256):LB%=E1-HB%*256          <070>
51340 PRINT#1,CHR$(LB%);CHR$(HB%);            <018>
51350 FOR I=E1 TO E2                          <145>
51360 :PRINT#1,CHR$(USR(I));                  <232>
51370 NEXT I:CLOSE 1                          <146>
51375 GOSUB 8000 :REM INDEX-SEQ-FILE          <115>
51380 PRINT"*** FERTIG ***":END               <196>
51500 REM ALS STRING SPEICHERN  ------        <221>
51501 IF LEN(A$)<MB(II)-MA(II)+1 THEN A$=A
      $+" ":GOTO 51501                        <221>
51510 FOR I=MA(II)TO MB(II)                   <090>
51515 :IF ASC(MID$(A$,I,1))=29 THEN POKE I
      Z,32:IZ=IZ+1:GOTO 51520                 <169>
51516 :POKE IZ,ASC(MID$(A$,I,1)):IZ=IZ+1      <240>
51520 NEXT I                                  <039>
51550 RETURN                                  <043>
51600 REM ALS VAL-WERT SPEICHERN ------       <009>
51610 POKE IZ,VAL(MID$(A$,MA(II),MB(II)))     <197>
51620 IZ=IZ+1                                 <158>
51630 RETURN                                  <123>
60000 REM MARKIEREN WIE DATIERT WIRD          <108>
60010 I$="":FOR I=1 TO LEN(A$)                <225>
60012 : IF ASC(MID$(A$,I,1))=29 THEN I$=I$
      +" ":GOTO 60015                         <149>
60014 : I$=I$+MID$(A$,I,1)                    <012>
60015 NEXT I                                  <152>
60019 FOR I=1 TO LEN(I$):M$(I)="0":NEXT       <123>
60020 IM=1:PRINT                              <131>
60025 PRINT I$:FOR I=1 TO LEN(I$):PRINT M$
      (I);:NEXT:PRINT                         <223>
60030 PRINT"{DOWN}EDIT{3SPACE}>";MID$(I$,I
      M,1);"<{4SPACE,RVSON}"IM"{RVOFF}"       <232>
60035 I=PEEK(203):IF I=64 THEN 60035          <110>
60040 IF I=2 AND PEEK(653)=0 AND IM<>LEN(I
      $)THEN IM=IM+1:GOTO 60070               <174>
60042 IF I=2 AND PEEK(653)=1 AND IM<>1 THE
      N IM=IM-1:GOTO 60070                    <119>
60045 IF I=56 THEN M$(IM)="1":GOTO 60070      <041>
60047 IF I=59 AND IG=0 THEN M$(IM)="2":GOT
      O 60070                                 <012>
60050 IF I=8 AND IG=0 THEN M$(IM)="3":GOTO
       60070                                  <164>
60055 IF I=35 THEN M$(IM)="0":GOTO 60070      <178>
60056 IF I=1 THEN 60100                       <013>
60060 GOTO 60035                              <183>
60070 PRINT"{UP,25SPACE,25LEFT,3UP}";:GOTO
       60025                                  <139>
60100 REM UMRECHNEN                           <252>
60102 FOR I=1 TO 3:MA(I)=0:MB(I)=0:NEXT I     <148>
60105 FOR II=1 TO 3:REM  PRINT"{RVSON}II={
      RVOFF}";II                              <194>
60110 :FOR I=1 TO LEN(I$):IF MA(II)=0 AND
      M$(I)=RIGHT$(STR$(II),1)THEN MA(II)=
      I                                       <059>
60112 :REM PRINT"M$(I)=";M$(I);"{2SPACE}RI
      GHT$"RIGHT$(STH$(II),1)                 <051>
60113 NEXT I                                  <250>
60115 :IF MA(II)=0 THEN 60150                 <245>
60120 :   MB(II)=MA(II)                       <124>
60125 :   MB(II)=MB(II)+1:REM PRINT,MB(II)    <069>
60130 :   IF M$(MB(II))<>RIGHT$(STR$(II),1
      )THEN MB(II)=MB(II)-1:GOTO 60150        <217>
60135 :   IF MB(II)=LEN(I$)THEN 60150         <074>
60140 :   GOTO 60125                          <248>
60150 NEXT II                                 <179>
60200 PRINT"{CLR,RVSON}";I$:PRINT"{RVOFF}W
      URDE IN FOLGENDE BEREICHE ZERLEGT:{D
      OWN}"                                   <143>
60210 FOR I=1 TO 3                            <108>
60215 IF MA(I)=0 THEN 60230                   <051>
60220 PRINT"BEREICH";I;":{SPACE,RVSON}"MID
      $(I$,MA(I),MB(I)-MA(I)+1)"{RVOFF}"      <126>
60230 NEXT I:PRINT                            <071>
60240 PRINT"BITTE GEBEN SIE NUN AN, OB DIE
       BEREICHE                               <037>
60245 PRINT"ALS{SPACE,RVSON}V{RVOFF}AL-WER
      T ODER ALS{SPACE,RVSON}S{RVOFF}TRING    <147>
60250 PRINT"GESPEICHERT WERDEN SOLLEN!{DOW
      N}"                                     <028>
60260 FOR I=1 TO 3                            <158>
60262 IF MA(I)=0 THEN 60290                   <228>
60265 PRINT"{RVSON}"MID$(I$,MA(I),MB(I)-MA
      (I)+1)"{RVOFF}"                         <229>
60270 POKE 198,0:INPUT"{RVSON}V{RVOFF,SPAC
      E}ODER{SPACE,RVSON}S{RVOFF}";II$        <045>
60275 IF II$<>"V"AND II$<>"S"THEN 60270       <095>
60280 TY%(I)=1:IF II$="S"THEN TY%(I)=2        <110>
60290 NEXT I                                  <173>
60300 RETURN                                  <157>
61000 CLOSE 1:OPEN 1,8,15                     <037>
61010 INPUT#1,A,B$,C,D                        <014>
61020 PRINT A,B$,C,D                          <126>
61030 CLOSE 1                                 <078>
```

**Listing 9. RAM-Editor (Schluß)**

RAM von 40960-65535 - also 25 KByte. Für den, der englische Textadventures programmieren will, ist das Problem damit auch schon gelöst. Wir beanspruchen für unsere Adventures jedoch RAM zur Unterbringung des deutschen Zeichensatzes. Trotzdem bleiben uns noch 12 KByte RAM am Stück - nämlich von 40960 bis 52223. Das ist immerhin genug Platz für zirka 480 Wörter - vorausgesetzt, wir halten an unserem bisherigen Wortschatzaufbau fest. Wenn Sie sich unseren Wortschatz einmal anschauen, so werden Sie schnell feststellen, daß eigentlich kein Verb länger als 16 Zeichen ist. Zum Speichern von Wortcode und Wortart genügen eigentlich auch 2 Byte (vorausgesetzt, es gibt maximal 255 verschiedene Wörter einer Wortart wie zum Beispiel Verben). So würden uns schon 18 Byte zum Speichern eines Wortes genügen. Wir würden dann 12000/18 = 682 Wörter im RAM unterbringen, womit unser erstes Ziel erreicht wäre.

## Der RAM-Editor

Bevor man allerdings einen Wortschatz innerhalb eines freien RAM-Bereiches verwalten will, muß man erst einmal einen Wortschatz ins RAM bringen. Wie man das am besten macht? Nun, die Worte einzeln per Hand ins RAM POKEn ist mit Sicherheit nicht das Gelbe vom Ei. Eine Idee wäre es vielleicht, einen Editor zu schreiben, der für diese Arbeit geeignet ist. Zusätzlich würde man dann natürlich noch einen Supersorter benötigen, der die Worte in diesem RAM-Bereich sortieren kann. Diese beiden Hilfsprogramme zu programmieren macht sehr viel Arbeit. Für uns gibt es hier eine Alternative, die außerdem sogar noch ihre Vorteile hat: Schließlich haben wir doch schon unseren relativen Datei-Editor und einen Supersorter, mit denen wir relative Dateien beziehungsweise relative Wortschätze prima in den Griff bekommen. Wir brauchen also nur ein Programm, das es uns ermöglicht, relative Dateien von der Diskette ins RAM zu schreiben. Wie diese RAM-Dateien dann verwaltet werden, soll uns an dieser Stelle noch nicht interessieren. Betrachten wir doch noch einmal einen kleinen Teil unserer Wortschatz-Datei:

```
1   abwaerts     2010
2   aufwaerts    2009
3   bedanken     1074
4   bedecken     1037
5   befestigen   1042
```

Dies sind die ersten fünf Sätze der relativen Wortschatz-Datei. Nehmen wir nun einmal an, diese fünf Sätze sollen ins RAM ab Speicherzelle 40960 geschrieben werden. Wir wissen, daß sich jeder Satz aus 25 Zeichen zusammensetzt:

1-2 das eigentliche Wort
22 die Wortzahl
23-25 der Wortcode

Nehmen wir außerdem an, daß wir zur Speicherung von Wortzahl und Wortcode jeweils nur 1 Byte benötigen (also maximal 255 Verben, 255 Objekte etc.), so können wir davon ausgehen, daß wir zur Speicherung eines Satzes der relativen Datei 21+2=23 Byte RAM benötigen. Für fünf Sätze sind dies 5 mal 23 = 115 Byte. Wir beanspruchen also den RAM-Speicher von 40960-41075. Bild 10 zeigt, wie die Sätze im RAM stehen.

Satz 1 steht jetzt also ab Speicherzelle 40960, Satz 2 ab 40983 und so weiter. In Adresse 40960+21 steht die Wortzahl, in 40960+22 der Wortcode des 1. Wortes. Entsprechendes gilt für alle anderen Sätze. Sie sehen schon, daß es gar nicht so schwer ist, im RAM ein relative-Dateien-ähnliches Verwaltungssystem aufzubauen. Bevor wir uns jedoch näher mit der Verwaltung von RAM-Dateien beschäftigen, brauchen wir erst einmal ein Programm (Listing 9), mit dem es möglich ist, relative Dateien kurz und schmerzlos in RAM-Dateien umzuwandeln.

Der RAM-Editor ist ein reines Hilfsprogramm. Ich verzichte daher auf eine Dokumentation - es gibt auch kaum eine sinnvolle Idee, um die man den Editor noch erweitern könnte.

Lassen Sie uns nun unsere relative Wortschatzdatei in eine RAM-Datei umwandeln: Nach dem Programmstart fordert der RAM-Editor Sie auf, den Namen der Datei anzugeben, von der Daten ins RAM geschrieben werden sollen. Geben Sie hier »Wortschatz« an, legen Sie die Diskette, auf der sich die Wortschatz-Datei befindet, ins Laufwerk und drücken Sie RETURN. Sodann müssen Anfang und Ende des Bereiches angegeben werden, der ins RAM übertragen werden soll. Wir wollen die gesamte Wortschatzdatei umwandeln: also Anfang = 1 und Ende = 197 (da wir 197 Wörter im Wortschatz haben). Nun müssen Anfang und Ende des RAM-Bereiches angegeben werden, in dem die Daten später stehen sollen. Um das Programm allgemein zu halten, müssen Sie dies

```
40960 ABWAERTS
40983 AUFWAERTS
41006 BEDANKEN
41029 BEDECKEN
41052 BEFESTIGEN
```

WORT
WZ
WC
Hier stehen Zeichen, die später als Zahl gelten

Bild 10. So stehen die Wörter im RAM

```
ram-editor v. 1. 5.          von michael nickles

ram-editor kann in freien ram-bereichen relative dateien
verwalten.
bitte name der relativen datei angeben, von der daten ins
ram geschrieben werden sollen (disk einlegen)!

filenam[illegible]ortschatz
bitte geben sie anfang und ende des bereiches an, der ins
ram geschrieben werden soll!

anfang? 1
ende  ? 197

bitte geben sie anfang und ende des ram-bereiches an, in
dem die daten abgelegt werden sollen! achtung - diese
angaben werden nicht ueberprueft!

anfang? 40960
ende? 45491

abwaerts               2010
1111111111111111111112333

edit   >0<    25

abwaerts               2010
wurde in folgende bereiche zerlegt:

bereich 1 : abwaerts
bereich 2 : 2
bereich 3 : 010

bitte geben sie nun an, ob die bereiche als val-wert oder
als string gespeichert werden sollen!

abwaerts
v oder s? s
2
v oder s? v
010
v oder s? v

bitte warten!

bitte diskette einlegen, auf der ram-bereich gespeichert
wird! (return
filename? wortschatz

bitte warten!
moechten sie einen index-file erstellen?
ab speicherstelle ? 40960
bis               ? 45491
schrittweite-step ? 23

bitte warten!
filename ? index
*** fertig ***
```

Disk mit relativer Datei einlegen und Filenamen angeben
Anfang und Ende des Bereiches angeben
RAM-Bereich von-bis angeben
Markieren
Angeben, ob Bereich als VAL-Wert oder String abgehandelt wird
Wortschatz als PRG-File speichern
Index erstellen

Bild 11. Beispiellauf des RAM-Editors

selbst ausrechnen. Wir wollen unseren Wortschatz ab Speicherstelle 40960 unterbringen. Warum der Editor nicht selbständig ermitteln kann, erkläre ich Ihnen gleich. Wir haben 197 Wörter und brauchen pro Wort 23 Byte RAM (wie wir bereits ermittelt haben). Es werden für den gesamten Wortschatz also 197 mal 23 = 4531 Byte benötigt. Das Ende des RAM-Bereiches errechnet sich also aus 40960 + 4531 = 45491. Jetzt wird der erste Satz der relativen Datei eingeblendet. Anhand der Cursortasten left-right und der Zifferntasten 1 bis 0 können Sie jetzt einzelne Satzteile markieren. In Bild 11 können Sie sehen, wie im Falle unserer Wortschatz-Datei markiert werden muß. Der Markierungsvorgang wird mit der RETURN-Taste abgeschlossen. Jetzt erscheint auf dem Bildschirm, wie der eben markierte Satz in einzelne Bereiche zerlegt wurde. Es können maximal 3 Bereiche markiert werden – in beliebiger Reihenfolge! Sie müssen nun zu jedem einzelnen Bereich angeben, ob er als VAL-Wert oder als String gespeichert werden soll. Bei String wird der Bereich einfach in ASC-Zahlen umgewandelt, die ins RAM gePOKEt werden. Bei VAL wird aus dem Bereich ein VAL-Wert ermittelt (der nicht größer als 255 sein darf – gegebenenfalls müssen Sie in High- und Low-Byte aufteilen), der dann einfach an die entsprechende Stelle im RAM gePOKEt wird. Sie müssen nun eine Zeitlang warten. Dann werden Sie aufgefordert, einen Filenamen anzugeben, unter dem die Wortschatz-RAM-Datei als PRG-File auf Diskette gespeichert wird. Bitte legen Sie hier eine leere Diskette ein und geben Sie als Filenamen »Wortschatz« an. Jetzt dauert's wieder eine Weile. Nun kann ein Index-File erstellt werden, was bei Wortschatz-Dateien in der Regel notwendig ist. Es müssen wieder Anfang und Ende des RAM-Bereiches angegeben werden und außerdem die Schrittweite – dies ist die Länge jedes Satzes im RAM – in unserem Fall 23 (Byte). Jetzt speichern Sie den erstellten Index unter dem Filenamen »Index« ab, und fertig ist die RAM-Wortschatzdatei. Dies alles hört sich viel komplizierter an, als es eigentlich ist. Bitte vergleichen Sie stets mit Bild 11, damit Sie beim Erstellen der RAM-Datei nichts falsch machen!

Die Wortschatz-Datei kann jetzt mittels LOAD»WORTSCHATZ«,8,1 geladen werden.

Der Index ist als SEQ-File gespeichert und kann folgendermaßen eingelesen werden:

```
5 OPEN3,4
10 OPEN1,8,2,"INDEX"
20 FORI=1TO26
30 :INPUT#1,X
40 :PRINT#3,CHR$(64+I)" AB ADRESSE"X
50 NEXTI
60 CLOSE1
70 CLOSE3
```

Wenn Sie alles richtig gemacht haben, dann müssen Sie

```
10 IF I=0 THEN I=1:LOAD"WORTSCHATZ",8,1        <067>
20 DIM BE$(25)                                  <194>
30 DIM IN(26):FOR I=1 TO 26:READ IN(I):NEX
   T I                                          <215>
40 DATA 40960,41006,0,41581,41650               <044>
50 DATA 42018,42110,42317,42501,0               <219>
60 DATA 42547,42685,0,42823,43007               <140>
70 DATA 43076,43237,43260,43605,44364           <159>
80 DATA 44502,44571,45031,0,0,45284             <175>
100 REM MODIFIZIERTE PEEK-FUNKTION --------
    ------------------------------------        <034>
101 :                                           <077>
110 DATA 32,247,183,120,169,52,133,1            <234>
120 DATA 160,0,177,20,168,169,55,133            <116>
130 DATA 1,88,76,162,179                        <215>
140 FOR I=704 TO 724:READ X:POKE I,X:S=S+X
    :NEXT I                                     <059>
150 POKE 785,192:POKE 786,2:REM USR-ZEIGER      <159>
160 IF S<>2325 THEN PRINT"FEHLER IN DATAS{
    SHIFT-SPACE}!":END                          <173>
200 GOSUB 50000                                 <212>
210 PRINT"VE="VE:PRINT"O1="O1:PRINT"O2="O2
    :PRINT"AD="AD:PRINT"RI="RI                  <051>
230 GOTO 200                                    <166>
50000 REM *************************             <093>
50001 REM *                       *             <009>
50002 REM *    WORT-PARSER 5.0    *             <041>
50003 REM *                       *             <011>
50004 REM *     <C> 1986 BEI      *             <086>
50005 REM *                       *             <013>
50006 REM *    MICHAEL NICKLES    *             <082>
50007 REM *                       *             <015>
50008 REM *************************             <101>
50010 REM BEFEHLSEINGABE ----------------
      -----------------------------------
      --                                        <076>
50011 :                                         <202>
50015 J1=1:J2=10:J3=198:J4=54276:J5=54277:
      J6=203:J7=3000:J8=2:J9=20:J0=54273        <111>
50016 K1=54272:K2=89:K3=129:K4=47:K5=57:K6
      =65:K7=90:K8=32:K9=34:K0=44               <068>
50017 PRINT                                     <078>
50019 POKE 54296,15:REM UND ABFRAGEN            <081>
50020 TT=.: REM INPUT-INTERRUPT-ZAEHLER         <076>
50030 FOR I=J1 TO J2:BE$(I)="":NEXT:POKE J
      3,.:BE$="": REM SATZ LOESCHEN             <058>
50040 PRINT"{YELLOW}>";:PRINT"@";               <238>
50050 POKE J4,.:POKE J5,.:GET X$:II=PEEK(J
      6):IF TT<J7 THEN 50060:REM TT=TT+1 !
      !!!                                       <010>
50055 FOR I=1 TO LEN(BE$)+J8:PRINT CHR$(J9
      );:NEXT:VE=0:RETURN                       <233>
50060 IF X$=""THEN 50050                        <223>
50061 POKE J5,J1:POKE J0,J2:POKE K1,K2:POK
      E J4,K3                                   <033>
50070 TT=0                                      <065>
50080 IF II=1 THEN IF BE$<>""THEN PRINT"{L
      EFT}.":GOTO 50200                         <158>
50085 I=ASC(X$):IF I>K4 AND I<K5 THEN 5010
      0                                         <214>
50090 I=ASC(X$):IF I<K6 OR I>K7 THEN IF I<
      >K8 AND I<>J9 AND I<>K9 AND I<>K0 TH
      EN 50050                                  <146>
50100 IF I=J9 AND LEN(BE$)=.THEN 50050          <103>
50110 IF I=J9 THEN PRINT"{2LEFT,2SPACE,2LE
      FT}@";:BE$=LEFT$(BE$,LEN(BE$)-J1):GO
      TO 50050                                  <072>
50120 BE$=BE$+CHR$(I):IF LEN(BE$)>78 THEN
      BE$=LEFT$(BE$,78):GOTO 50050              <168>
50125 IF I=K9 THEN I=64                         <168>
50130 PRINT"{LEFT}";CHR$(I);"@";:GOTO 5005
      0                                         <007>
50200 REM CODIERUNG ----------------------
      -----------------------------------
      --                                        <020>
50201 :                                         <138>
50250 REM WORT IM RAM SUCHEN *******            <127>
50254 I$=BE$                                    <243>
50255 IM=ASC(LEFT$(I$,1)):I=IN(IM-64)           <124>
50262 REM BEGIN LOOP            <<<             <230>
50264 :IF IM<>USR(I)THEN 50280                  <120>
50265 :FOR I1=1 TO LEN(I$):PRINT MID$(I$,I
      1,1),CHR$(USR(I+I1-1))                    <205>
50270 : IF ASC(MID$(I$,I1,1))<>USR(I+I1-1)
      THEN I1=LEN(I$):GOTO 50276                <157>
50273 : IF I1=LEN(I$)AND USR(I+21)=3 THEN
      OB=USR(I+22):OM=OB:PRINT"OBJEKT="OB:
      END                                       <050>
50274 : IF I1=LEN(I$)AND USR(I+21)=2 THEN
      RI=USR(I+22):PRINT"RICHTUNG="RI:END       <241>
50275 : IF I1=LEN(I$)AND USR(I+21)=1 THEN
      VE=USR(I+22):PRINT"VERB="VE:END           <012>
50276 :NEXT I1                                  <163>
50278 I=I+23:GOTO 50264                         <079>
50280 REM END LOOP              <<<             <002>
50290 PRINT"ICH KENNE DAS WORT "BE$" NICHT
       !":END                                   <076>
```

**Listing 10. Grundgerüst vom Parser 5.0**

jetzt folgende Wertetabelle (bei Ausgabe auf Drucker) erhalten:

```
A AB ADRESSE 40960
B AB ADRESSE 41006
C AB ADRESSE 0
D AB ADRESSE 41581
E AB ADRESSE 41650
F AB ADRESSE 42018
G AB ADRESSE 42110
H AB ADRESSE 42317
I AB ADRESSE 42501
J AB ADRESSE 0
K AB ADRESSE 42547
L AB ADRESSE 42685
M AB ADRESSE 0
N AB ADRESSE 42823
O AB ADRESSE 43007
P AB ADRESSE 43076
Q AB ADRESSE 43237
R AB ADRESSE 43260
S AB ADRESSE 43605
T AB ADRESSE 44364
U AB ADRESSE 44502
V AB ADRESSE 44571
W AB ADRESSE 45031
X AB ADRESSE 0
Y AB ADRESSE 0
Z AB ADRESSE 45284
```

Jetzt steht der Programmierung von Parser 5.0, dem Superparser, nichts mehr im Wege. Im Prinzip ist es sogar ein leichtes, unseren Parser 4.0 so zu modifizieren, daß er statt mit einer relativen Datei mit einer RAM-Datei arbeitet. Dies ist jedoch nicht empfehlenswert, da es bei RAM-Dateien wesentlich effektivere Verwaltungsmöglichkeiten gibt, die allerdings etwas komplizierter als bei der relativen Disk-Verwaltung verlaufen.

## Leider nicht komplett

Um Ihnen eine Enttäuschung zu bewahren, muß ich Sie bereits an dieser Stelle darauf aufmerksam machen, daß Sie auf diesen Seiten keineswegs einen kompletten Parser 5.0 finden. Es gibt einfach zu viele Möglichkeiten bei der Verwaltung von RAM-Wortschätzen, von denen Sie am besten selbst entscheiden, welche Ihnen am meisten behagt, nachdem ich Ihnen verschiedene vorgestellt habe.

Bitte geben Sie jedoch zunächst einmal Listing 10 ein.

Dieses Programm ist, wie bereits gesagt, noch kein kompletter Parser. Es soll Ihnen vielmehr als Grundgerüst eines Parsers 5.0 nach eigenen Vorstellungen dienen.

Dokumentation zum Listing:

10 Laden des Wortschatzes von Diskette ins RAM.
30 Indextabelle IN (1)–IN(26) initialisieren
40–80 Index-DATAs. Hier wurde nichts weiter gemacht, als der Index-File ausgelesen und in DATAs untergebracht.
100–160 Hier steckt eine für Parser 5.0 grundlegende Erweiterung: Eine modifizierte PEEK-Funktion, die es erlaubt, die Speicherzellen unter dem Basic-ROM auszulesen. Sie wissen mit Sicherheit bereits, daß beim POKEn in diesen Bereich immer ins RAM, aber beim normalen PEEKen immer der Speicherinhalt vom ROM ausgegeben wird.

Die erweiterte PEEK-Funktion wird mittels USR(X) aufgerufen. Wenn Sie zum Beispiel in Speicherzelle 60000 den Wert 1 schreiben mit POKE 60000,1, so erhalten Sie bei PRINT PEEK (60000) einen Wert, der nicht unbedingt der 1 entspricht. PRINT USR(60000) zeigt jedoch den Wert vom RAM: 1.

Diese Erweiterung ist für alle diejenigen unentbehrlich, die Parser 5.0 ausschließlich in Basic programmieren wollen.

200 Aufruf von Parser 5.0
50010–50130 Erweiterte INPUT-Routine. Diesmal mit schönem Eingabe-Ton.
50200 Das Herzstück von Parser 5.0 beginnt hier.
–50290 Er erfolgt hier die Suche nach dem Wort. Das Programm ist sehr einfach gehalten. Es können nur Befehlssätze eingegeben beziehungsweise codiert werden, die nicht mehr als ein Wort enthalten. Sie können dies anhand des bisher erworbenen Wissens jedoch leicht ändern.
50254 Das Suchwort I$ soll im RAM gesucht werden.
50255 In der Variablen IM wird der ASC-Wert des ersten Buchstabens vom Suchwort gespeichert. Anhand der Index-Tabelle wird jetzt ermittelt, ab welcher Speicherzelle (=I) die Wörter stehen, deren erster Buchstabe den ASC-Wert IM hat.
50262–50280 Hier steht die eigentliche Suchroutine.
50264 Wenn IM nicht gleich USR(I) ist, so heißt dies, daß an der RAM-Stelle I ein Wort steht, dessen erster Buchstabe nicht mit dem Suchwort I$ übereinstimmt. Der Suchvorgang wird dann mittels GOTO 50290 abgebrochen, und es erfolgt die altbekannte Fehlermeldung »ICH KENNE DIESES WORT NICHT!«.
50265 Der gesamte Suchbegriff I$ wird vom ersten bis zum letzten Zeichen durchlaufen. Es wird dabei verglichen, ob ASC(MID$(I$,1,1)–ASC(MID$(I$,LEN(I$), 1) gleich USR(I–1) – USR(I–1+LEN(I$)) ist. Es werden also der Reihe nach die einzelnen Buchstaben des Suchwortes mit den entsprechenden Speicherzellen im RAM verglichen, von denen vermutet wird, daß sie dem Buchstaben entsprechen.
50270 Wird bei irgendeinem Buchstaben des Wortes eine Nichtübereinstimmung mit dem Wert im RAM festgestellt, so wird der Suchvorgang abgebrochen mit GOTO 50276, und der Zeiger I wird auf das nächste Wort im RAM gesetzt, dessen erster Buchstabe IM ist: I=I+23 (23, da jedes Wort beziehungsweise jeder Satz 23 Byte lang ist).
50273–50275 Wurde bis zum letzten Buchstabenvergleich keine Nichtübereinstimmung festgestellt, so wurde das Suchwort im RAM gefunden. Anhand der Wortart WA=USR(I+21) wird dann die entsprechende Wortzahl WA=USR(I+22) ermittelt. Das Wort ist somit codiert.

Wenn Sie das Programm einmal laufen lassen, werden Sie überrascht sein, wie schnell die Wörter im RAM gefunden werden. Schon das Grundmodell von Parser 5.0 ist weitaus schneller als Parser 4.0 und Parser 4.1. Allerdings läßt sich die Verarbeitungsgeschwindigkeit noch erhöhen:

Man kann zum Beispiel eine index-binäre Suchroutine wie bei Parser 4.0 einbauen. Ergänzt man diese index-binäre Methode zu einer index-Feld-binären Suchroutine, so erhält man einen Basic-Parser, wie man ihn sich schneller kaum noch vorstellen kann.

Ein index-Feld-binärer Parser 5.0 kann an Geschwindigkeit nur noch mit Maschinensprache-Programmierung übertrof-

| BASIC-RAM | OBJEKTE | VIDEO | VERBEN etc. | D.T. z. SATZ | |
|---|---|---|---|---|---|

40960 52223 53248 57344 61440 65535

**Bild 12. Speicherbelegung des C64, wenn die wichtigsten Routinen eingebaut sind.**

fen werden. Allerdings kann man in puncto Wortschatz-Organisation noch einiges verbessern, wie die folgenden Beispiele zeigen:

## Verbesserte Wortschatz-Organisation

Wie wir bereits festgestellt haben, besteht ein Wortschatz aus Verben, Objekten, Adjektiven und Richtungsangaben und AUS-Wörtern. Als Wortlänge haben wir uns bisher maximal 21 Zeichen reserviert. Sind aber wirklich 21 Zeichen nötig? – Mit Sicherheit nicht. Ein Verb zum Beispiel ist in 99% aller Fälle an 6 Buchstaben zu erkennen. Das heißt, kennt man die ersten 6 Buchstaben eines Verbs, so weiß man mit großer Wahrscheinlichkeit, welches Verb gemeint ist.

Betrachten wir doch einmal die ersten 6 Sätze unserer Wortschatz-Datei im Falle einer Verbkürzung auf 6 Zeichen:

```
abwaer 2010
aufwae 2009
bedank 1074
bedeck 1073
befest 1042
beklei 1065
belohn 1046
benutz 1063
beraub 1058
berueh 1017
```

Bei einer Umwandlung zur RAM-Datei benötigt man in diesem Fall nur noch 8 Byte – das heißt, man bringt fast dreimal soviele Verben unter wie bisher!

Es ist auch ein leichtes, eine neue Wortschatzdatei zu erstellen, bei der die Verben auf 6 Zeichen Länge begrenzt werden. Sie müssen beim Markieren mit dem RAM-Editor lediglich nur die ersten 6 Zeichen markieren. Bei der Suchroutine ist jetzt allerdings zu beachten, daß im Falle einer Verb-Suche nur noch die ersten 6 Buchstaben des Suchwortes im RAM gesucht werden. Und zwar linksbündig, wie es sich für die Verbsuche gehört. Analog für die Verben kann man auch die Adjektive und Richtungsangaben abhandeln. Bei den AUS-Wörtern, also den Wörtern, die eigentlich zur Codierung gar nicht benötigt werden (zum Beispiel der, die, das, den etc.), gilt dies im großen und ganzen auch.

Etwas schwieriger ist es mit den Objekten. Objekte können sehr lange Namen haben, wie zum Beispiel »Fernsehapparat« oder »Zeitmaschine«.

Behalten wir als Musterbeispiel einmal den Objektnamen »Zeitmaschine« bei. Objekte haben wir bei Parser 4.0 rechtsbündig gesucht (damit sowohl TÜR als auch HAUSTÜR verstanden wird etc.). Die letzten 6 Buchstaben von »Zeitmaschine« lauten »schine«, und daraus läßt sich wahrlich nicht erkennen, daß es sich um eine »Zeitmaschine« handelt. Würde man für die Objekte 10 Buchstaben reservieren, so käme man schon besser zurecht. Bei relativen Dateien muß jeder Satz die gleiche Länge haben, und es kann immer nur eine relative Datei gleichzeitig geöffnet sein. Deshalb waren wir bei Parser 4.0 und 4.1 auch gezwungen, eine große Wortlänge für die Verben zu reservieren, obwohl diese eigentlich nur für Objekte nötig ist. Eine Alternative wäre es gewesen, Objekte und die anderen Wortarten zusammen, in zwei getrennten relativen Dateien unterzubringen. Das Öffnen und Schließen von relativen Dateien wäre dann jedoch unvermeidbar gewesen. Das hätte Parser 4.0 stark verlangsamt, weil das Öffnen von relativen Dateien viel Zeit in Anspruch nimmt. Also eine miserable oder keine Alternative. Anders bei RAM-Dateien. Von ihnen können beliebig viele gleichzeitig geöffnet werden beziehungsweise sind immer offen. Es bietet sich also an, für die Objekte einen anders organisierten RAM-Bereich herzunehmen. Dies ist auch der Grund, warum ich unseren bisherigen Wortschatz frei von Objekten, nur aus Verben bestehend, gelassen habe. Ich bin mir ziemlich sicher, daß Sie bei Ihren eigenen Adventures zu den vorhandenen 197 Verben und Richtungsangaben kaum noch mehr als 50 andere finden. Wenn Sie sich also Platz für 400 Wörter mit 6 Zeichen Länge (250 Verben + 150 Adjektive) RAM reservieren wollen, so brauchen Sie 400 mal 8 = 3200 Byte Speicherplatz. Für Verben, Richtungsangaben und Adjektive und eventuell auch noch AUS-Wörter können wir das freie RAM von 53248 bis 57343 (= Platz für 511 Wörter mit 8 Zeichen Länge – mit 2 Zeichen gerechnet für Wortart und Wortcode). Jetzt haben wir noch freies RAM von 40960 bis 52222 – also genug Platz für 938 Objekte bei Objektlänge = 11 Zeichen + Wortcode (auf die Wortart können wir verzichten, da wir wissen, daß sich in diesem Bereich nur Objekte befinden).



938 + 511, das bringt einen Wortschatz von 1449 Wörtern. Was soll also das unsinnige Infocom-Getue um den fantastischen 1000er-Wortschatz. 1449 Wörter, das sind mehr Wörter, als der englische Grundwortschatz hat, mit dem sich immerhin schon 70 % aller Dinge in Englisch formulieren lassen, sind garantiert genug für jedes nur erdenkliche Adventure, das auf einem C 64 geschrieben werden soll. Jetzt liegt es nur noch an Ihnen, Parser 5.0 zu programmieren. Wenn Sie das gesamte Parser-Technik-Kapitel aufmerksam durchgearbeitet haben, dürfte Ihnen diese Programmierung keine Schwierigkeiten mehr bereiten (wobei Sie noch nicht einmal in Maschinensprache programmieren müssen)!

Bild 12 zeigt, wie es jetzt im RAM des C 64 aussieht:

Der freie RAM-Bereich ist nahezu voll ausgenutzt mit dem Wortschatz und dem deutschen Zeichensatz. Besser kann man das RAM im Bereich der Adventureprogrammierung garantiert nicht mehr ausnutzen. Sogar die Grafikroutine (vergleiche Kapitel Grafikprogrammierung) läßt sich in dieses RAM-Schema noch integrieren – es geht dann nur Basic-RAM verloren sowie 1 KByte des Objekt-RAMs, wobei aber immer noch Platz für genügend Objekte vorhanden ist. Den RAM-Bereich von 61440 bis 65535 lasse ich Ihnen für eigene Ideen offen. Den Bereich von 1024 bis 2048, der durch das Verschieben des Bildschirms freigeworden ist, werde ich allerdings noch beanspruchen.

Wir haben es also geschafft – wir haben viele Basic-Routinen und Parser kennengelernt und das RAM des C 64 optimal mit Wortschatz, Grafikfenster und deutschem Zeichensatz belegt. Es steht also nur noch aus, das ganze Wissen zusammenzupacken und daraus Adventures zu programmieren. Wie das geht, erfahren Sie im anschließenden Kapitel.

(M. Nickles/bs)

# Super-Grafik für Adventures

**In diesem Artikel wollen wir wichtige Tips für diejenigen bringen, die ihr Adventure grafisch aufmöbeln wollen. Sie können dann also nicht nur professionelle Text- sondern auch Grafikadventures programmieren.**

Wer heute ein außergewöhnlich gutes Adventure schreiben will, muß sich früher oder später die Frage stellen, ob es sinnvoll ist, das Spiel mit Grafik zu versehen oder nicht. Letztendlich ist diese Entscheidung auch darüber ausschlaggebend, ob zirka 8 KByte RAM für eine Bit-Map oder für einen riesigen Wortschatz ausgenutzt werden. Die letzten Abenteuerspiele, die auf den Markt gekommen sind, haben auch eine traurige Tatsache immer wieder bestätigt: Die Spiele, die mit fantastischer Grafik ausgestattet sind, lassen immer am Wortschatz schwer zu wünschen übrig. Ich denke hier zum Beispiel an das Spiel Asylum: Die Grafik ist zwar überwältigend, aber die Spiellust geht spätestens dann verloren, wenn man zum zehntenmal die Antwort »You can not do that« erhalten hat und niemals eine detaillierte Erklärung, warum etwas nicht geht. Unglücklicherweise enthält das Spiel auch noch eine Dia-Show-Funktion, die es ermöglicht, zu Beginn des Spiels alle Bilder anzuschauen. Dadurch wird schließlich auch noch der letzte Funke Spiellust zerstört – nämlich die Überraschung beim Entdecken eines neuen Bildes. Wir wollen uns zunächst jedoch nicht mit Speicherplatzproblemen im Bereich HiRes-Bilder/Wortschatz beschäftigen, sondern vielmehr die folgende Frage klären: Wie macht man fantastische HiRes-Bilder für Grafikadventures?

Jeder von Ihnen hat bestimmt schon einmal mit Malprogrammen wie zum Beispiel Koala Painter oder Paint Magic etc. gearbeitet. Begabte Grafiker kommen damit oft auch zu guten Resultaten. Allerdings ist das »per Hand Zeichnen« unheimlich zeitaufwendig. Zugegeben, es gibt zahlreiche Zeichenprogramme, die das Arbeiten mit Grafiktabletts, Trackball, Joystick, Maus und Lightpen unterstützen, aber alle diese Methoden erfordern unheimliches Geschick zum Zeichnen – und das haben nur sehr wenige Programmierer (zu denen auch ich leider nicht gehöre). Die Lösung allen Übels heißt schließlich: Video Digitizer. Video-Digitizer sind Geräte, mit denen es möglich ist, Bilder von einer Kamera beziehungsweise einem Videogerät in den Speicher eines Computers einzulesen (sprich digitalisieren). Diese Geräte waren bis vor kurzem sehr teuer (zirka 600 Mark), sind jetzt aber schon relativ erschwinglich. Gebrauchtgeräte gibt es schon ab 200 Mark.

Leider ist es mit dem Video-Digitizer allein noch lange nicht getan. Neben einer Computeranlage benötigt man auch noch eine Video-Kamera. Für so eine Schwarzweiß-Kamera muß man dann noch einmal zirka 600 Mark auf den Tisch legen. Es lohnt sich jedoch in den seltensten Fällen, eine solche Kamera zu kaufen, die lediglich zum Arbeiten mit dem Digitizer verwendbar ist.

Eine andere Möglichkeit ist es, einen Videorecorder an den Digitizer anzuschließen. Dies hat jedoch nur dann einen Sinn, wenn der Recorder über ein ausgezeichnetes Standbild verfügt. An einen Videorecorder mit Standbild zu gelangen beziehungsweise sich einen von Freunden auszuleihen, dürfte kein allzu großes Problem sein. Anders die Kamera. Gute Videokameras sind im Moment noch dermaßen teuer, daß sie nur in wenigen Haushalten zu finden sind. Es gibt jedoch zahlreiche Videotheken und Video-Geschäfte, in denen man Kameras (und natürlich auch Recorder) ausleihen kann. Wer Glück hat, findet bestimmt ein Geschäft in dem man eine Videoanlage für weniger als 50 DM ein Wochenende lang ausleihen kann. Alles, was Sie dann noch benötigen, sind Bildvorlagen zum Digitalisieren. Sind die Bilder erst einmal im Computer, ist es ein leichtes, sie nach eigenen Wünschen zu manipulieren, umzufärben und schließlich ins eigene Adventure einzubauen. Man benötigt also noch ein gutes Malprogramm, um die digitalisierten Bilder vernünftig bearbeiten zu können. Leider hat jedes Malprogramm Vorteile und Nachteile – und die Bilder der einzelnen Programme sind natürlich absolut inkompatibel untereinander.

Viele Digitalisierer speichern im Koala-Painter-Format ab. Als Zeichenprogramm würde ich aber Paint Magic empfehlen. Deswegen nun ein kleines Listing zum Verwandeln von Koala- in Paint-Magic-Bilder (Listing 1). Das kurze Listing erklärt sich von selbst.

In manchen Fällen ist es nötig, daß die Farbeinstellungen nachträglich per Hand korrigiert werden müssen.

## Paint Magic Diashow

Im folgenden wird erklärt, wie Sie Paint-Magic-Bilder innerhalb von Basic-Programmen laden und darstellen können. Die Paint-Magic-Bilder können übrigens auch direkt geladen werden und mittels RUN betrachtet werden, da vor jedem Bild ein kleines Programm zur Bilddarstellung mit gespeichert wird.

```
10 REM ********************************     <148>
20 REM *   BILDUMSCHREIBE-ROUTINE     *     <039>
30 REM *.                             *     <079>
40 REM *    KOALA - PAINT MAGIC       *     <120>
50 REM *                              *     <099>
55 REM ********************************     <193>
60 IF A=1 THEN 130                          <217>
70 PRINT"{CLR,YELLOW,DOWN,RVSON,2SPACE}BIL
   DUMSCHREIBEROUTINE{17SPACE,RVOFF}"       <196>
80 PRINT"{2SPACE}QUELLDISKETTE EINLEGEN ,
   DANN BILD-{4SPACE,DOWN,SPACE}KENNZIFFER
    EINGEBEN !"                             <123>
90 INPUT"{2DOWN}";K$                        <108>
100 IF K$<"A"OR K$>"Z"THEN 50               <116>
110 POKE 55,0:POKE 56,96                    <009>
120 A=1:LOAD"?PIC "+K$+"*",8,1              <237>
130 :                                       <106>
140 :                                       <116>
150 PRINT"{DOWN,RVSON,2SPACE}BILDUMSCHREIB
    EROUTINE{17SPACE,RVOFF}"                <070>
160 PRINT" ZIELDISK EINLEGEN , DANN GEWUEN
    SCHTEN{3SPACE,DOWN}NAMEN EINGEBEN !"    <207>
170 INPUT"{DOWN,SPACE}";N$                  <063>
180 OPEN 1,8,1,N$+",P,W"                    <064>
190 PRINT#1,CHR$(142);CHR$(63);             <096>
200 FOR I=16270 TO 16383:PRINT#1,CHR$(0);:
    NEXT                                    <155>
210 FOR I=24576 TO 32767:PRINT#1,CHR$(PEEK
    (I));:NEXT                              <025>
220 FOR I=32576 TO 33599:PRINT#1,CHR$(PEEK
    (I));:NEXT                              <084>
230 CLOSE 1                                 <241>
```

**Listing 1. Der Bildumwandler**

Und so werden die Bilder innerhalb von Programmen geladen und dargestellt:

Zunächst müssen sie in einer neuen Form gespeichert werden. Dazu:

1. Bild laden.
2. Bild mit »RUN« einschalten.
3. Bild mittels RUN-STOP/RESTORE ausschalten.
4. POKE 24565,96 im Direktmodus eingeben.
5. Den Bereich von hex 4000 - 6400 (=Bild) mit einem Monitor speichern.

Das so gespeicherte Paint-Magic-Bild kann jetzt innerhalb eines Programmes absolut geladen werden (,8,1) und mittels SYS 24513 dargestellt werden.

Um das Bild wieder auszuschalten, sind folgende POKEs notwendig:

```
POKE 53272,21 : POKE 56576,151 : POKE 53265,27
POKE 53270,200
```

So lassen sich ganz einfach Dia-Shows herstellen.

## Fenstertechnik auf dem C 64

Im folgenden zeige ich Ihnen, wie Sie HiRes-Bilder in Ihre eigenen Adventures einbauen können. Die Idee besteht darin, den Bildschirm des C 64 in eine Grafik und in ein Textfenster zu unterteilen.

Bild 1 zeigt, wie der Bildschirm unterteilt wird. In der oberen Hälfte wird ein Multicolor-HiRes-Fenster mit der Auflösung 96 mal 160 Bildpunkte dargestellt.

Die 13. Textzeile bleibt als Trennung leer. Die unteren 12 Zeilen sind Text-Bildschirm.

Die Routine ist speziell für Adventures gedacht, die auf relativer Dateien-Basis und mit einem deutschen Zeichensatz arbeiten. Das MSE-Listing zur Fensterroutine finden Sie in Listing 2.

Wie bereits gesagt, arbeitet die Routine auch dann, wenn der deutsche Zeichensatz verwendet wird. Das Ladeprogramm, mit dem zu Beginn eines Adventures die Fenster-

MULTICOLOR-MODE
AUFLÖSUNG = 96 x 160 Pixel
Grafik-Fenster (12 x 40 Zeichen)
LEERZEILE
Textmode
12 Zeilen à 40 Zeichen
Textfenster (12 x 40 Zeichen)

**Bild 1. Aufteilung des Bildschirms**

```
programm : fenster            9c00 9d4b
----------------------------------------

9c00 : 00 4c 0a 9c 4c ea 9c 4c   64
9c08 : 16 9d 78 a9 00 8d 00 9c   e6
9c10 : ae c3 9c 20 c4 9c a9 01   a5
9c18 : 8d 12 d0 ad 11 d0 29 7f   d3
9c20 : 8d 11 d0 a9 3f 8d 14 03   56
9c28 : a9 9c 8d 15 03 a9 7f 8d   bc
9c30 : 0d dc a9 01 8d 1a d0 ad   7e
9c38 : 19 d0 8d 19 d0 58 60 ad   ec
9c40 : 00 9c f0 25 ce 00 9c a9   22
9c48 : 01 8d 12 d0 ea ea ea ea   36
9c50 : ea ea ea ea ea ea ea ea   4f
9c58 : ea ea ea ea ea 20 a8 9c   5b
9c60 : ad 19 d0 8d 19 d0 4c 31   2b
9c68 : ea ee 00 9c a9 91 8d 12   de
9c70 : d0 20 7d 9c ad 19 d0 8d   45
9c78 : 19 d0 4c 81 ea a9 01 8d   58
9c80 : 00 dd a9 3b 8d 11 d0 a9   39
9c88 : d8 8d 16 d0 a9 80 8d 18   cb
9c90 : d0 ad c2 9c 8d 21 d0 a0   e1
9c98 : 28 a9 00 99 df d9 ad c3   d3
9ca0 : 9c 99 b7 d9 88 d0 f2 60   cd
9ca8 : a9 00 8d 00 dd a9 1b 8d   67
9cb0 : 11 d0 a9 c8 8d 16 d0 a9   cd
9cb8 : 3a 8d 18 d0 a9 00 8d 21   ec
9cc0 : d0 60 0c 00 a9 00 85 fb   6c
9cc8 : a8 a9 d8 85 fc 8a 91 fb   8e
9cd0 : 18 a5 fb 69 01 85 fb a5   5e
9cd8 : fc 69 00 85 fc a5 fb c9   ba
9ce0 : 08 d0 ea a5 fc c9 da d0   eb
9ce8 : e4 60 78 20 a8 9c a2 07   26
9cf0 : 20 c4 9c a9 01 8d 12 d0   35
9cf8 : a9 0a 8d 14 03 a9 9d 8d   9b
9d00 : 15 03 58 ad 19 d0 8d 19   e3
9d08 : d0 60 ad 19 d0 8d 19 d0   16
9d10 : 4c 31 ea 41 20 42 8d 13   48
9d18 : 9d a9 42 8d 15 9d 20 36   f7
9d20 : 9d a9 56 8d 15 9d 20 36   04
9d28 : 9d ad ff 9f 8d c2 9c ad   4c
9d30 : fe 9f 8d c3 9c 60 a2 08   41
9d38 : a0 01 20 ba ff a9 03 a2   57
9d40 : 13 a0 9d 20 bd ff a9 00   91
9d48 : 4c d5 ff 24 3d 4c 45 46   db
```

**Listing 2. Die Fenster-Routine**

Zero-page | RAM | BASIC-RAM | BITMAP | RAM | Video | RAM | D. Z. Satz | RAM
0, 1024, 2048, 32768, Fenster-routine, 41984, 52223, 53248, 57344, 61440, 65535, 39136, 40960

**Bild 2. Die aktuelle Speicherbelegung**

Routine geladen und der deutsche Zeichensatz geladen und aktiviert wird, finden Sie in Listing 3.

In Zeile 30 wird der Zeiger auf die Speicher-Obergrenze verschoben. Dies ist notwendig, um die Routine, die teils den Basic-RAM beansprucht, vor Überschreiben durch Basic-Programme zu schützen.

Bild 2 zeigt, wie es jetzt im Speicher des C 64 aussieht. Die Fenster-Routine nimmt schon ziemlich viel RAM in Anspruch:

| | |
|---|---|
| 32768–39135 | Unterbringung der Bit-Map |
| 39136–40959 | Hier steht die eigentliche Fenster-Routine |
| 40960–41983 | Farbinformationen für das HiRes-Bild |

Jetzt stehen nur noch 30717 KByte Basic-RAM zur Verfügung. Die schräg gestrichelten Bereiche sind die noch freien RAM-Bereiche. Wer Adventures mit Grafik-Fenster programmieren will, muß also ziemlich hart kalkulieren. Aufgrund der Adventure-Programmiertechniken, die Sie bisher in diesem Kurs kennengelernt haben, können Sie den Verlust des Fenster-RAMs durch geschickte Programmierung wieder ausgleichen.

Hier noch ein Tip für alle ausgefuchsten Programmierer, die kein Byte ungenutzt sein lassen wollen: Der RAM-Bereich, der für Bit-Map und Farbinfo benutzt wird, wird eigentlich gar nicht ganz beansprucht, da unsere Fensterbilder nur eine Auflösung von 96 mal 160 haben. Das heißt, in den RAM-Bereichen von 36929–39135 und 41480–41982 können noch kleine Maschinenroutinen untergebracht werden. Und so wird mit der Fensterroutine gearbeitet:

```
10 IF A=1 THEN 50                                  <070>
20 IF A=2 THEN 60                                  <241>
30 POKE 55,0:POKE 56,128                           <199>
40 CLR:A=1:LOAD"D.Z.SATZ",8,1                      <186>
50 A=2:LOAD"FENSTER",8,1                           <053>
60 POKE 648,204:POKE 53272,56:POKE 56576,0         <192>
70 PRINT"{CLR}"                                    <058>
```

**Listing 3. Lader für die Fenster-Routine**

```
5 PRINT"{CLR,RVSON}BILDUMSCHREIBEROUTINE{R
  VOFF}"                                           <221>
10 REM BILDUMSCHREIBER PAINT MAGIC                 <014>
20 REM                                             <082>
30 REM VON WOLFGANG BORGSMUELLER                   <255>
40 REM                                             <102>
45 IF A=1 THEN 500                                 <201>
50 POKE 55,144:POKE 56,48                          <055>
60 CLR:A=1                                         <053>
65 PRINT"BITTE QUELLDISK EINLEGEN UND NAME
   DES{3SPACE}PAINT-MAGIC BILDES EINGEBEN
   !"                                              <163>
66 INPUT"NAME ";F$                                 <060>
70 LOAD F$,8,1                                     <226>
80 END                                             <082>
500 PRINT"ZIELDISK EINLEGEN!":POKE 198,0           <085>
510 GET A$:IF A$=""GOTO 510                        <095>
511 PRINT"BITTE GEBEN SIE DEN KENNBUCHSTAB
    EN DES{2SPACE}FENSTERBILDES EIN!               <111>
512 INPUT C$                                       <118>
520 OPEN 1,8,1,LEFT$(C$,1)+" B,P,W"                <101>
525 PRINT#1,CHR$(0);CHR$(128);                     <013>
530 FOR I=16384 TO 21184                           <001>
540 PRINT#1,CHR$(PEEK(I));:NEXT                    <113>
550 CLOSE 1                                        <053>
555 INPUT"HG-FARBE DES BILDES";HG                  <247>
556 INPUT"FARBE 4 DES BILDES";F4                   <041>
560 OPEN 1,8,1,LEFT$(C$,1)+" V,P,W"                <221>
565 PRINT"BILDNAME: ";C$                           <219>
570 PRINT#1,CHR$(254);CHR$(159);CHR$(F4);C
    HR$(HG);                                       <058>
580 FOR I=24576 TO 25176                           <215>
590 PRINT#1,CHR$(PEEK(I));:NEXT                    <163>
600 CLOSE 1                                        <103>
```

**Listing 4. Wandler Paint Magic → Fensterformat**

Bei ausgeschaltetem Fenster steht der ganze Bildschirm als Textfenster zur Verfügung. Mittels SYS 39937 wird das Fenster eingeschaltet beziehungsweise eingeblendet. SYS 39940 schaltet das Fenster wieder aus. Der Text beziehungsweise Bildschirminhalt, der sich unter dem eingeblendeten Fenster befand, ist jetzt wieder sichtbar. Allerdings ist dieser Text einfarbig. Die Farbe (0–15), die der Text nach Ausschalten des HiRes-Fensters annimmt, steht in Speicherzelle 40175. Mittels POKE 40175,FARBE kann also variiert werden. Auch Text, der unter das eingeschaltete Fenster gescrollt wird, ist nach Ausschalten des Fensters wieder sichtbar. Das Fenster ist völlig Interrupt-gesteuert – deshalb flimmert die unterste Zeile manchmal ein wenig, was jedoch kaum stört.

Durch die Interrupt-Programmierung ist es jetzt auch nicht mehr so einfach, zwischen dem Groß/klein- und Groß/grafik-Zeichensatz umzuschalten. Hier muß auch wieder gePOKEt werden. Und zwar POKE 40120,58 um den Groß/klein-Zeichensatz und POKE 40120,56 um den Groß/grafik-Zeichensatz einzuschalten. Jetzt müssen Sie natürlich noch wissen, wie man Fenstergrafiken lädt. Hierzu dient ein kleines Programm, das Paint-Magic ins Fensterformat umwandelt. Sie benötigen also Paint-Magic-Bilder, wobei das spätere Fensterbild die oberen 12 Zeilen (beziehungsweise 96 vertikale Bildpunkte) des Paint-Magic-Bildes einnimmt.

Das Programm in Listing 4 funktioniert ganz einfach: Zuerst muß die Diskette, auf der sich das Paint-Magic-Bild befindet, eingelegt werden. Nach dem Laden wird das Programm auf einer Zieldiskette gespeichert. Wichtig ist dabei, daß der Filename des Fensterbildes nur ein einziges Zeichen lang sein darf. Außerdem müssen die Hintergrundfarbe (1. Farbe) des Bildes sowie die 4. Farbe angegeben werden – also mit Hilfe von Paint-Magic lesen und notieren. Auf einer Diskette finden 30 Bilder im Fensterformat Platz. Jetzt müssen Sie nur noch wissen, wie Bilder mit Hilfe der Fensterroutine geladen werden:

```
POKE 780, ASC("Zeichen") : SYS 39943
```

Für »Zeichen« muß der Filename des Bildes (der aus nur einem Zeichen bestehen darf) eingesetzt werden.

## Probleme mit der Fensterroutine

Bei eingeschaltetem Fenster kommt es zu Schwierigkeiten bei Disk-Operationen (zum Beispiel Laden von relativen Files): der Bildschirm beginnt dann wild zu flackern. Dies liegt daran, daß Diskettenoperationen störend auf programmierte Video-Interrupts wirken. Vielleicht kann man dies sogar als einen Fehler im C 64-Betriebssystem bezeichnen. Es ist möglich, Fensterroutinen beziehungsweise Interrupts zu programmieren, bei denen Diskettenoperationen nicht stören. Dazu muß jedoch das Betriebssystem vom ROM ins RAM kopiert und dort verändert werden. Für diese Lösung habe ich mich jedoch nicht entschieden, weil dann zuviel RAM-Speicher verlorengegangen wäre. Man muß vor Diskettenoperationen das Fenster aus- und erst später wieder einschalten.

Zusammenfassung

| | |
|---|---|
| Fenster einschalten: | SYS 39937 |
| Fenster ausschalten: | SYS 39940 |
| Bild laden: | POKE 780,ASC ("Zeichen"):SYS 39943 |
| Zeichenfarbe hinter Fenster: | POKE 40175,FARBE |
| Groß/klein-Zeichensatz einschalten: | POKE 40120,58 |
| Groß/grafik-Zeichensatz einschalten: | POKE 40120,56 |

Viel Spaß beim Fensterln!

# Adventure-Programmiertechnik

**Es ist soweit. Im nun folgenden Abschnitt erfahren Sie endlich, wie man sie programmiert: Super-Adventures, wie es sie bis jetzt noch nicht gibt. Sie erhalten alle notwendigen Grundlagen, um professionelle Adventures zu erstellen.**

In meinem ersten Adventure-Kurs habe ich bereits ausführlich beschrieben, wie man eine Adventure-Idee aufgreift, diese ausarbeitet, Pläne anfertigt etc. Wir wollen uns deshalb hier gar nicht mehr erst lange mit solchen Grundlagen aufhalten, sondern gleich zur Sache kommen. Auf den nächsten Seiten programmieren wir gemeinsam die ersten Fortsetzungsszenen von Gordon Saga II. An dieser Stelle muß ich auch gleich noch einmal ein paar Worte zur Gordon Saga I verlieren. Ich möchte mich bei allen begeisterten Adventurern bedanken, die im Spiel mit mühevoller Kleinstarbeit meine Telefonnummer herausgefunden und mich dann angerufen haben. Ich erhielt zahlreiche Anrufe aus allen Teilen Deutschlands, Österreichs und der Schweiz. Bedanken möchte ich mich vor allem deshalb, weil ich bisher keinen einzigen bösen Anruf erhielt. Einige teilten mir sogar mit, daß das Spiel Fehler enthält und deshalb nicht lösbar sei. Es stimmt, es sind einige Fehler enthalten, aber ich kann versichern, daß diese Fehler in keinster Weise zur Unlösbarkeit beziehungsweise zum Absturz des Spiels führen: es sind lediglich kleine Logikfehler, wie sie bei der Programmierung eines komplexen Adventures unvermeidbar sind. An dieser Stelle sei auch die »Incredible Four« gegrüßt, die Gordon Saga I geknackt und beim Titelbild das HAPPY-SOFTWARE-Emblem durch UNHAPPY SOFTWARE ersetzt haben (PS: Wenn ich euch erwische, dreh ich euch den Hals um!). »Wo bleibt Gordon Saga, Teil II« haben mich viele gefragt. Die Programmierung von Teil II habe ich kurz vor Abschluß abgebrochen. Dies hat seinen guten Grund: Teil II ist einfach um Längen besser als Teil I in puncto Text, Grafik und Parser. Wenn jemand, der Gordon Saga I kennt, von Gordon Saga II hört, so vergleicht er unwillkürlich den zweiten mit dem ersten Teil – und dies wäre nicht gerade die beste Werbung.

Als kleinen Vorgeschmack möchte ich gemeinsam mit Ihnen nun die ersten Szenen von Gordon Saga II programmieren. Dazu ist notwendig, daß Sie wissen, wie Gordon Saga I ausgeht. Am Spielende treibt der Spieler mit einem Floß auf einem stillen unterirdischen See in einem geheimnisvollen Gebirge. Plötzlich entsteht im See ein riesiger Strudel, der den Spieler samt Floß in die Tiefe reißt – die Pforte zu einer geheimnisvollen Welt hat sich aufgetan ...

Und dies sind die ersten Szenen von Gordon Saga II: Der Spieler wacht in einer Glaskuppel am Grunde des Sees wieder auf. Von der Glaskuppel aus führt ein Tunnel in die Dunkelheit. Durch diesen Tunnel gelangt man in eine Kristallgrotte ...

Den Lageplan für diesen Teil des Spieles zeigt Bild 1.

Wir haben 4 Räume: Glaskuppel, Tunnel, Kristallgrotte und Felsplateau. Außerdem ist bei jedem Raum aufgezählt, welche Objekte sich in ihm befinden. Von Raum 3 gelangt man nicht so ohne weiteres zu Raum 4, was durch die gestrichelte Linie angedeutet ist – man braucht irgendwie das Schwert.

Wie fängt man jetzt an zu programmieren?

Nun, man muß sich erst einmal überlegen, wie man die einzelnen Programmteile – also Basic-Programm, Zeichensatzlader, Grafikbilder, relative Dateien etc. auf einer Disk unterbringt. Am besten, man entscheidet sich, eine Diskette beidseitig zu verwenden.

Auf Seite 1 bringt man das Grundprogramm, den deutschen Zeichensatz, eventuell Maschinenprogramme und Grafikbilder unter.

Das heißt, außer den Grafikbildern kommt auf Seite 1 alles, was einmal in den Rechner geladen wird und dann dort bleibt. Auf Seite 2 kommt die relative Datei, auf der alle Texte, Kommentare und Raumbeschreibungen gespeichert sind. Außerdem wird auf Seite 2 die relative Wortschatzdatei untergebracht. In der Entwicklungsphase eines Adventures eignet sich Parser 4.0 zweifellos am besten – man kann leicht neue Wörter einfügen etc. Erst ganz am Ende, wenn alle Spielszenen fertig programmiert sind – das Adventure also fertig ist –, sollte man endgültig auf Parser 5.0 umrüsten.

Nehmen Sie sich jetzt bitte eine gute Diskette her, die für beidseitiges Bespielen geeignet ist, und formatieren Sie beide Seiten. Dann nehmen Sie die Disk-Utility her und ändern die IDs der Disketten so, daß der ID der ersten Seite »1side«, der der zweiten Seite »2side« lautet. Dies ist notwendig, damit später innerhalb des Programmes festgestellt werden kann, ob die richtige Disk-Seite eingelegt ist. Jetzt brauchen wir den Relativ-Datei-Editor. Mit ihm eröffnen wir auf der 2. Diskseite eine relative Datei mit folgenden Angaben:

Satzlänge 41
Anzahl Sätze 3000
Filename »TEXTE«

Jetzt müssen Sie auf die zweite Diskseite noch die relative Wortschatzdatei überspielen. Dies geht nicht so ohne weiteres, da es nur sehr wenige Kopierprogramme gibt, die relative Files einzeln kopieren können. Sie können jedoch mit dem Editor eine neue Wortschatzdatei anlegen und dann mittels VERSETZEN-Befehl (siehe auch Anleitung zum Editor) die alte Wortschatzdatei in die neu angelegte übertragen. Eine andere Möglichkeit besteht darin, einfach die ganze Diskettenseite, auf der sich die Wortschatzdatei befindet, mit einem Diskettenkopierprogramm auf die zweite Seite unserer jetzigen Arbeitsdiskette zu kopieren und dann mittels Scratch

4 FELSPLATEAU
3 KRISTALLGROTTE
KRISTALLE
QUADER, STATUE
AUSGANG, SCHWERT
SCHWERT
2 TUNNEL
1 GLASKUPPEL
KUPPEL; SAND

**Bild 1. Der Lageplan von Gordonsaga II (Ausschnitt)**

und Validate den Rest zu löschen. Diskseite 2 muß jetzt so aussehen:

```
0   "gordon saga ii     "2side
492 "texte"              rel
32  "wortschatz"         rel
140 blocks free.
```

Jetzt können wir mit der Arbeit beginnen.
Der erste Schritt besteht darin, den Wortschatz und die Wörter zu erweitern, die für die aktuelle Spielszene, die sich in der Regel aus einigen Räumen zusammensetzt, benötigt werden. Man sollte nie zu viele Räume auf einmal behandeln.

## Wortschatz erweitern

Denken Sie sich immer eine kleine Szene aus, programmieren Sie diese und arbeiten Sie erst dann weiter, wenn diese Szene einwandfrei läuft. Durch dieses Szene-für-Szene-Programmieren hält sich die Fehlersuche in engen Grenzen, und man kann sicher sein, daß das Adventure bei der Fertigstellung nur wenige Logikfehler enthält – da jede Szene für sich ausgiebig getestet wurde. Wie erweitert man den Wortschatz? Ganz einfach. Für unsere jetzige Szene müssen wir den Wortschatz um folgende Objekte ergänzen:

```
Glaskuppel       1
Sand             2
Tunnel           3
Kristallgrotte   4
Quader           5
Statue           6
Schwert          7
Zettel           8
Amulett          9
Grube            10
Oeffnung         11
Felsplateau      12
Kristalle        13
```

Der Wortschatz wird ganz einfach erweitert, indem Sie mit dem relativen Datei-Editor die neuen Wörter ans Ende des bisherigen Wortschatzes anhängen (ab Satz Nr. 198) und dann die gesamte Datei neu sortieren. Hängen Sie bitte folgende Sätze an:

```
198  glaskuppel      3001
199  sand            3002
200  tunnel          3003
201  kristallgrotte  3004
202  quader          3005
203  statue          3006
204  schwert         3007
205  zettel          3008
206  amulett         3009
207  grube           3010
208  oeffnung        3011
209  felsplateau     3012
210  kristalle       3013
```

Jetzt muß der Wortschatz neu sortiert werden. Beachten Sie bitte, daß mit der Sortierroutine des relativen Datei-Editors nur Dateien sortiert werden können, die nicht mehr als 300 Sätze haben. Übersteigt die Anzahl der Worte im Wortschatz diese Zahl, so müssen Sie zum Sortieren des Wortschatzes den Super-Sorter hernehmen, bei dem kein Limit besteht.

Nachdem der Wortschatz erweitert ist, können wir uns dem Text des Adventures zuwenden. Textadventures zeichnen sich besonders durch lange, beschreibende Texte aus. Nicht umsonst arbeiten viele Softwarehäuser bei der Erstellung ihrer Adventures mit Schriftstellern zusammen. Alleine Adventures schreiben ist langweilig. Auch sind Programmierer meistens keine guten Texter. Vielleicht tun Sie sich hier am besten mit einem Freund zusammen, der gerne schreibt. Zu zweit ein Spiel aushecken macht schließlich auch viel mehr Spaß.

Anschließend finden Sie den Text zu den ersten Szenen von Gordon Saga II. Bitte schreiben Sie ihn in die entsprechenden Sätze der Texte-Datei:

```
 1 Als Sie aus Ihrer Bewußlosigkeit
 2 erwachen, finden Sie sich auf einem
 3 weichen weißen Sandboden liegend
 4 wieder.
 5 Augenblicklich erinnern Sie sich an die
 6 letzten Ereignisse vor Ihrer
 7 Bewußtlosigkeit .. In Ihren Gedanken
 8 setzt sich das Abbild der riesigen
 9 Seeschlange zusammen, die Sie mit Hilfe
10 einer goldenen Armbrust erlegt hatten.
11 Ihre folgende Erinnerung gilt dem
12 plötzlich aufgekommenen Strudel, durch
13 den Ihr Floß samt Ihnen in die Tiefe
14 gerissen worden war.
15 Obwohl es Ihnen noch immer unbegreiflich
16 ist, wie Sie all dies überleben konnten,
17 machen Sie sich daran, Ihre jetzige
18 Umgebung näher zu untersuchen .....
19
20 GORDON SAGA II. Die Rückkehr des Helden
21 Ein Abenteuerspiel von Michael Nickles.
22 Parsersystem 4.0 - Copyright 1985
23 Auszug für Kurs im 64er-Sonderheft.        *
24 Glaskuppel
25 Sie befinden sich innerhalb einer
26 Glaskuppel am Grunde eines Sees.
27 Außerhalb der Kuppel ist das Wasser so
28 dunkel, daß man nichts erkennen kann.
29 Der weiße Sand am Boden strahlt ein
30 schwaches Licht aus, das die Kuppel
31 erhellt. Ein kleiner Tunnel führt von
32 hier aus hinaus in die endlose
33 Dunkelheit.                                 *
34 Tunnel
35 Sie befinden sich in einem zwei Meter
36 hohen und etwa drei Meter breiten
37 Tunnel. Die Wände sind aus einem Glas
38 angefertigt, das etwas trüber als das
39 Glas, aus dem die Kuppel besteht, ist.
40 Der Tunnel verläuft in nördliche
41 Richtung. Weit entfernt in nördlicher
42 Richtung sehen Sie ein helles Licht.        *
43 Kristallgrotte.
44 Dies ist eine riesige Grotte, deren Wände
45 mit Kristallen, die in allen Farben des
46 Lichtspektrums glitzern, besetzt sind.
47 Der Boden besteht aus glatten, strahlend
48 weißen Marmorplatten. Trotz zahlreicher
49 Unebenheiten ähnelt diese Grotte dem
50 Inneren einer Halbkugel mit einem Radius
51 von ca. 100 Metern. Der einzige Ausgang
52 von hier ist ein in südlicher Richtung
53 verlaufender Tunnel.                        *
54 Felsplateau
55 Dieses Felsplateau außerhalb der Grotte
56 bietet Ihnen eine fantastische Aussicht.
```

```
 57 Sie blicken über ein weites Land hinweg.
 58 das in Dunkelheit gehüllt ist. Am Himmel
 59 sehen Sie eine schwarze Scheibe, deren
 60 Ränder orangerot leuchten - es scheint,
 61 als sei die Dunkelheit durch eine
 62 andauernde Sonnenfinsternis bedingt.
 63 Etwa 300 Meter unterhalb des Plateaus
 64 befindet sich ein Wald mit einer Ihnen
 65 unbekannten Flora.
 66 All diese eindrucksvollen Anblicke
 67 verstärken in Ihnen das Gefühl, daß Sie
 68 sich in einer fremden Welt befinden.
 69 In der östlichen Gebirgswand ist ein
 70 Zugang zu einer Grotte.                 *
 71 Sie graben in dem leuchtenden Sand.     *
 72 Es ist sinnlos, noch tiefer zu graben.  *
 73 Nachdem Sie ein etwa 50 cm tiefes Loch
 74 in den Sand gegraben haben, finden Sie
 75 einen Zettel und ein Amulett.           *
 76 Im Sandboden ist eine ca. 50 cm tiefe
 77 Grube ausgehoben - es scheint so, als ob
 78 hier vor kurzem jemand gegraben hat!    *
 79 Auf dem Zettel steht etwas geschrieben! *
 80 "Supermike grüßt Captain Future,
 81 Section 8, Pink, Brutallica, Gommel,
 82 GCS (ich weiß, daß ihr genau 35 km
 83 Luftlinie von mir entfernt wohnt -
 84 Paßt bloß auf!!!), und natürlich alle
 85 Freunde, RAMBO und alle Girls, die mich
 86 besonders gut kennen!!"                 *
 87 Das Amulett ist aus massivem Gold
 88 angefertigt, mit schwarzen Edelsteinen
 89 besetzt und hängt an einer goldenen
 90 Halskette.                              *
 91 Die Grube ist völlig leer.              *
 92 Die Glaskuppel ist so groß, daß sich
 93 etwa 10 ausgewachsene Männer darin
 94 aufhalten können.                       *
 95 Der Sand ist sehr grobkörnig und
 96 leuchtet seltsamerweise.                *
 97 Die Kristalle glitzern in allen Farben
 98 des Lichtspektrums.                     *
 99 Die Kristalle sind fest in die Wände
100 eingelassen und lassen sich auch mit
101 größten Anstrengungen nicht entfernen.  *
102 Auf dem Marmorboden, exakt im Zentrum
103 des Raumes, steht ein goldener Quader,
104 von dem eine ca. 3 Meter hohe Statue
105 emporragt.                              *
106 Nur wenige Meter von der Statue entfernt
107 steckt ein goldenes Schwert zu etwa
108 einem Drittel seiner Gesamtlänge im
109 Marmorboden.                            *
110 Das Schwert ist fest im Marmorboden
111 verankert und läßt sich um keinen
112 Millimeter herausziehen.                *
113 Die Statue gibt die groben Umrisse eines
114 jungen Kriegers wieder. Er hat lange
115 Haare, ist von athletischer Körperstatur
116 und trägt eine Rüstung. Sie fühlen sich
117 von der Statue in einer seltsamen Art
118 und Weise angezogen.                    *
119 Als Sie die Statue mit Ihrer rechten
120 Hand berühren, durchfährt augenblicklich
121 eine Lähmung Ihren ganzen Körper und Sie
122 sind nicht mehr in der Lage, Ihren Arm
123 zurückzuziehen. Unmittelbar vor Ihren
124 Augen beginnt die Statue plötzlich ihr
125 Äußeres zu verändern.....
126 Die Lähmung läßt schließlich nach und
127 Sie stürzen benommen zu Boden.
128 Nachdem Sie den Schock überwunden haben,
129 stellen Sie fest, daß nicht nur die
130 Statue ihr Äußeres verändert hat -
131 während die Statue Ihre Form angenommen
132 hat, haben Sie den Körper des jungen
133 Kriegers erhalten, den die Statue noch
134 vor kurzem verkörpert hat.              *
135 Die Statue gibt jetzt die Umrisse und
136 Gesichtszüge eines Ihnen wohlvertrauten
137 Menschenkörpers wieder - ein Körper, der
138 noch vor kurzem der ihrige war!         *
139 Mit der unglaublichen Kraft, die in
140 Ihrem neuen Körper zu stecken scheint,
141 ziehen Sie das Schwert ohne jegliche
142 Anstrengung aus dem Marmorboden.
143 Kaum, daß Sie es in Ihren Händen halten,
144 beginnt der gesamte Raum um Sie herum
145 leicht zu beben und das Schwert strahlt
146 in gleißendem Licht.
147 Aus dem Nichts ertönt plötzlich eine
148 Ihnen altbekannte Stimme. "In Dir
149 erfüllt sich die alte Prophezeiung!
150 Geh, GORDON, die Macht ist mit Dir!"
151 Als die letzten Worte des Uralten
152 gerade verklingen, tut sich in der
153 westlichen Wand der Grotte eine Öffnung
154 auf und gibt den Ausgang zu einem
155 Felsplateau frei.
156 Sie stehen verwundert da und fragen sich
157 immer wieder, warum gerade Sie der
158 Auserwählte sind und zu welchem Zweck
159 der Uralte Sie in diese fremde Welt
160 geholt hat.....                         *
161 In der westlichen Wand der Grotte ist
162 eine Öffnung, die den Ausgang zu einem
163 Felsplateau ermöglicht.                 *
164 die Glaskuppel
165 der Sand
166 der Tunnel
167 die Kristallgrotte
168 der Quader
169 die Statue
170 das Schwert
171 der Zettel
172 das Amulett
173 die Grube
174 die Öffnung
175 das Felsplateau
176 die Kristalle
```

Sie haben sicherlich bereits bemerkt, wie Adventuretexte in einer relativen Datei untergebracht werden: Jede Raumbeschreibung oder jeder Kommentar wird mit einem Sternchen abgeschlossen. Nur die Objektnamen selbst stehen allein in einer Zeile, den Geschlechtsartikel vorangestellt.

Damit ist auch der Text für die ersten Szenen auf Disk vorhanden. Jetzt benötigen wir ein Programmiergerüst, um das herum wir unser Adventure aufbauen. Dieses Gerüst finden Sie in Listing 1.

Dieses Gerüst soll Ihnen als Grundprogramm für jedes Adventure dienen. In ihm sind nahezu alle Routinen enthalten, die für die Programmierung beziehungsweise Entwicklung eines Textadventures notwendig sind.

```
0 REM ***************************           <109>
1 REM GORDON SAGA II - 1985 BY SM           <188>
2 REM ***************************           <111>
3 IF I=0 THEN I=1:LOAD"D.Z.SATZ",8,1        <076>
4 POKE 648,204:POKE 53272,56:POKE 56576,0:
  PRINT"{CLR}":REM D.T.SATZ AKTIVIEREN      <144>
5 OPEN 15,8,15:PRINT"{CLR,YELLOW}";CHR$(14
  );                                        <204>
6 SIDE=2:GOSUB 53200:PRINT"{CLR}";          <081>
15 GOSUB 60000: REM TABELLEN EINLESEN       <100>
20 DIM WI(100): REM ZIMMER-MERKER           <253>
30 SC=0:MO=0 : REM SCORE/VERSUCH            <065>
40 SA=1:GOSUB 53600
               : REM TITELTEXT              <004>
50 ZN=1:GOSUB 54000
               : REM BESCHREIB. 1.RAUM      <067>
100 REM START ----------------------------
    -------------------------------------- <206>
101 :                                       <077>
110 IF SB<>0 THEN SA=SB:SB=0:GOSUB 53600:R
    EM KOMMENTAR AUSGEBEN                   <133>
120 GOSUB 50000           :REM BEFEHLSEING
    ABE                                     <150>
130 :                                       <106>
300 REM GEH-ROUTINE ------------------      <218>
310 IF RI=0 THEN 400                        <021>
320 I=MAP(ZN,RI)                            <017>
330 IF I=0 THEN PRINT"DIESE RICHTUNG IST N
    ICHT MÖGLICH!":GOTO 400                 <119>
340 ZN=I:GOSUB 54000                        <186>
400 REM SCHAU/UNTERSUCHE -------------      <129>
410 IF VE<>11 THEN 500                      <034>
415 IF O1=0 THEN GOSUB 54000:GOTO 500       <137>
420 SA=OS(O1):GOSUB 53400:GOSUB 53500:OB$=
    "D"+RIGHT$(I$,LEN(I$)-1)                <154>
425 IF OB(O1)=-1 THEN 430                   <144>
430 IF OB(O1)<>ZN THEN X$=OB$+" IST HIER N
    ICHT ZU SEHEN!":GOSUB 53900:GOTO 500    <023>
440 IF OU(O1)=0 THEN X$=OB$+" HAT KEINE BE
    SONDEREN MERKMALE.":GOSUB 53900         <127>
450 IF OU(O1)THEN SA=OU(O1):GOSUB 53800     <153>
500 REM NIMM -------------------------      <042>
505 IF VE<>12 THEN 600                      <134>
506 IF SP=1 THEN 550:REM NIMM ALLES         <071>
507 SA=OS(O1):GOSUB 53400:GOSUB 53500:OB$=
    "D"+RIGHT$(I$,LEN(I$)-1)                <241>
510 IF O1=0 THEN PRINT"IHR SATZ ENTHÄLT KE
    IN OBJEKT !":GOTO 600                   <250>
514 :IF OB(O1)=-1 THEN PRINT"DAS HABEN SIE
     BEREITS !":GOTO 600                    <095>
515 :IF OB(O1)<>ZN THEN X$=OB$+" IST HIER
    NICHT ZU SEHEN!":GOSUB 53900:GOTO 600   <245>
516 IF OA(O1)=0 THEN 600                    <052>
520 :IF OA(O1)=1 THEN PRINT"BIZARR!":GOTO
    600                                     <065>
525 :IF OA(O1)=2 THEN X$=OB$+" IST VIEL ZU
     SCHWER ZUM TRAGEN!":GOSUB 53900:GOTO
    600                                     <121>
530 :IF OA(O1)=3 THEN PRINT"GENOMMEN.":OB(
    O1)=-1:GOTO 600                         <035>
550 :                                       <018>
554 PRINT:FOR I=1 TO OZ                     <149>
555 :IF OB(I)<>ZN THEN 580                  <246>
556 SA=OS(I):GOSUB 53400:GOSUB 53500:OB$="
    D"+RIGHT$(I$,LEN(I$)-1)                 <063>
560 :I$=OB$+": "                            <230>
565 IF SQ=1 AND I=O1 THEN 580:REM AUSSER    <106>
570 :IF OA(I)=1 THEN PRINT I$;"BIZARR !"    <133>
575 :IF OA(I)=2 THEN PRINT I$;"ZU SCHWER Z
    UM TRAGEN !"                            <131>
577 :IF OA(I)=3 THEN PRINT I$;"GENOMMEN.":
    OB(I)=-1                                <206>
580 NEXT I                                  <156>
600 REM INVENTUR -----------------          <178>
601 IF VE<>13 THEN 700                      <237>
605 PRINT"{DOWN}SIE HABEN:{SPACE,UP}"       <244>
610 IC=0:FOR I=1 TO OZ                      <250>
611 SA=OS(I):GOSUB 53400:GOSUB 53500:OB$="
    D"+RIGHT$(I$,LEN(I$)-1):I$=OB$          <194>
612 :IF LEFT$(I$,3)="DER"THEN J$=RIGHT$(I$
    ,LEN(I$)-3):I$="DEN"+J$                 <143>
615 :IF OB(I)=-1 THEN PRINT"{11RIGHT}";I$:
    IC=1                                    <139>
620 NEXT I:IF IC=0 THEN PRINT"{11RIGHT}BOC
    K AUF BIER !"                           <155>
700 REM VERLIERE -----------------          <100>
701 IF VE<>14 THEN 800                      <086>
702 IF SP=1 THEN 750: REM VERLIER ALLES     <156>
703 IF O1=0 THEN PRINT"IHR SATZ ENTHÄLT KE
    IN OBJEKT!":GOTO 800                    <062>
705 IF OB(O1)<>-1 THEN PRINT"{DOWN}SIE HAB
    EN DAS NICHT !":GOTO 800                <063>
715 OB(O1)=ZN:PRINT"OK.":GOTO 800           <137>
750 IC=0:FOR I=1 TO OZ                      <134>
755 :IF SQ=1 AND I=O1 THEN 780:REM AUSSER   <198>
760 SA=OS(I):GOSUB 53400:GOSUB 53500:OB$="
    D"+RIGHT$(I$,LEN(I$)-1):I$=OB$+": "     <115>
770 :IF OB(I)=-1 THEN OB(I)=ZN:PRINT I$;"O
    K.":IC=1                                <078>
780 NEXT I:IF IC=0 AND SQ=1 THEN PRINT"WIE
     SOLL DAS GEHEN ?"                      <053>
782 IF IC=0 AND SQ=0 THEN PRINT"SIE HABEN
    DOCH GARNICHTS !"                       <175>
800 :                                       <119>
2000 REM RAUMSPEZIFISCHE ACTION ----------
     -------------------------------------
     -                                      <056>
50000 REM ****************************      <227>
50001 REM *                          *      <009>
50002 REM *     WORT-PARSER 4.0      *      <169>
50003 REM *                          *      <011>
50004 REM *      <C> 1986 BEI        *      <086>
50005 REM *                          *      <013>
50006 REM *     MICHAEL NICKLES      *      <082>
50007 REM *                          *      <015>
50008 REM ****************************      <235>
50010 REM BEFEHLSSATZEINGABE -------------
      ------------------------------------
      --                                    <069>
50011 ZW=0:PRINT:GOSUB 53000:MO=MO+1:REM T
      ITELLINE AUSGEBEN                     <222>
50012 IF UD>0 THEN 50500:REM UND            <171>
50015 SL=77: REM BEFEHLSSATZLAENGE          <048>
50020 BE$="":POKE 198,0:PRINT"{YELLOW}>@";  <026>
50030 GET X$:IF PEEK(203)=1 THEN 50120      <226>
50040 IF X$=""THEN 50030                    <171>
50050 IF LEN(BE$)=0 AND ASC(X$)=20 THEN 50
      030                                   <148>
50060 I=ASC(X$):IF I<32 OR I>133 AND I<159
       THEN IF I<>20 THEN 50030             <131>
50070 IF LEN(BE$)=SL AND I<>20 THEN 50030   <128>
50080 BE$=BE$+X$                            <031>
50090 PRINT CHR$(20);X$;"@";                <108>
50100 IF I=20 THEN BE$=LEFT$(BE$,LEN(BE$)-
      2):GOTO 50030                         <101>
50110 GOTO 50030                            <001>
50120 PRINT"{LEFT,SPACE}"                   <227>
50400 REM SATZZEICHEN SUCHEN -------------
      ------------------------------------
      --                                    <103>
50401 :                                     <082>
50415 SN=0                                  <104>
50420 IF RIGHT$(BE$,1)=" "THEN BE$=LEFT$(B
      E$,LEN(BE$)-1):GOTO 50420             <233>
50430 I$=RIGHT$(BE$,1)                      <180>
50440 IF I$="."OR I$="!"THEN SN=0:BE$=LEFT
      $(BE$,LEN(BE$)-1):GOTO 50420          <010>
50450 IF I$="?"THEN SN=1:BE$=LEFT$(BE$,LEN
      (BE$)-1):GOTO 50420                   <144>
50500 REM BE$-ZEIGER SETZEN --------------
      ------------------------------------
      --                                    <199>
50501 :                                     <184>
50502 IF UD=0 THEN VE=0:RI=0:O1=0:O2=0:SP=
      0:SQ=0                                <157>
50504 IF UD>0 THEN RI=0:GOTO 50530          <199>
50505 IF LEFT$(BE$,1)=" "THEN BE$=RIGHT$(B
      E$,LEN(BE$)-1):GOTO 50505             <132>
50510 BE$=BE$+" "                           <232>
50520 ZA=1 :REM ZEIGER A SETZEN             <058>
50530 ZB=ZA+1                               <180>
50540 IF MID$(BE$,ZB,1)=""THEN UD=0:CLOSE
      1:OPEN 1,8,3,"TEXTE":RETURN           <170>
50550 IF MID$(BE$,ZB,1)<>" "THEN ZB=ZB+1:G
      OTO 50550                             <175>
50590 SU$=MID$(BE$,ZA,ZB-ZA)                <222>
```

**Listing 1. Ein vollständiges »Adventure-Programmier-Gerüst«**

```
50600 REM SU$ VORBEHANDLUNG --------------
      ------------------------------------
      --                                  <130>
50601 :                                   <028>
50610 IF SU$="UND"THEN UD=1:GOTO 50630    <093>
50620 IF SU$=","THEN UD=2:GOTO 50630      <061>
50625 GOTO 50670                          <201>
50630 ZA=ZB+1                             <080>
50640 IF MID$(BE$,ZA,1)=""THEN PRINT"UND W
      AS ?":GOTO 50530                    <153>
50650 IF MID$(BE$,ZA,1)=" "THEN ZA=ZA+1:GO
      TO 50650                            <242>
50660 CLOSE 1:OPEN 1,8,3,"TEXTE":RETURN   <119>
50670 IF SU$="IHN"OR SU$="SIE"OR SU$="ES"T
      HEN O1=OM:GOTO 51300                <069>
50680 IF SU$="ALLES"THEN SP=1:GOTO 51300  <029>
50690 IF SU$="AUSSER"THEN SQ=1:GOTO 51300 <012>
51000 REM BINAERE SUCHROUTINE ------------
      ------------------------------------
      --                                  <113>
51001 :                                   <176>
51002 CLOSE 1:OPEN 1,8,3,"WORTSCHATZ"     <031>
51007 REM ANFANG UND ENDE DES SUCHBEREICHE
      S ERMITTELN                         <117>
51008 I=ASC(LEFT$(SU$,1))                 <176>
51009 IF I-65<0 OR I-65>25 THEN GOSUB 5150
      0:GOTO 50000                        <134>
51010 AN=IN(I-65):REM ANFANG DES SUCHBEREI
      CHES                                <164>
51012 IZ=64                               <200>
51013 IF IN(I-IZ)=0 THEN IZ=IZ-1:GOTO 5101
      3                                   <086>
51015 EN=IN(I-IZ)-1:REM ENDE DES SUCHBEREI
      CHES                                <070>
51016 IF AN=0 THEN GOSUB 51500:UD=0:GOTO 5
      0000                                <249>
51020 SZ=197:N=INT(LOG(EN-AN+1)/LOG(2))+1
                 :REM MAXIMAL-FORMEL      <121>
51030 SA=AN-1+(2↑N)/2:REM MITTE DER GES.DA
      T                                   <073>
51040 GOSUB 52100:REM A$ LESEN            <234>
51050 N=N-1      :REM 1.ABFRAGE           <045>
51060 :                                   <235>
51070 REM SU$ MIT A$ VERGLEICHEN ------   <042>
51080 : IF LEN(SU$)<3 THEN 51110          <140>
51090 : IF WA=1 AND SU$=LEFT$(A$,LEN(SU$))
       THEN VE=WC:GOTO 51300              <157>
51100 : IF WA<>3 OR SU$<>RIGHT$(A$,LEN(SU$
      ))THEN 51110                        <161>
51102 :  IF UD=1 OR UD=2 THEN UD=3:O1=0:O2
      =0                                  <042>
51104 :   IF O1=0 THEN O1=WC:OM=O1:GOTO 51
      300                                 <133>
51105 :   IF O2=0 THEN O2=WC:OM=O1:GOTO 51
      300                                 <170>
51110 : IF WA=2 AND SU$=A$
       THEN RI=WC:GOTO 51300              <140>
51115 : IF WA=5 AND SU$=A$
       THEN AD=WC:GOTO 51300              <247>
51120 : IF WA=1 AND SU$=A$
       THEN VE=WC:GOTO 51300              <017>
51125 : IF WA=4 AND SU$=A$
       THEN 51300                         <204>
51130 : IF SU$<A$THEN SA=SA-(2↑(N-1)):GOSU
      B 52100:GOTO 51160                  <015>
51140 : IF SA+(2↑(N-1))>SA+EN-AN AND N>-1
      THEN N=N-1:GOTO 51140:REM ''UEBERLAU
      F''                                 <204>
51150 : SA=SA+(2↑(N-1)):GOSUB 52100:REM ''
      HALBIEREN''                         <141>
51160 :                                   <079>
51170 :N=N-1     :REM NAECHST. HALBIEREN  <068>
51180 :IF N<0 THEN GOSUB 51500:UD=0:GOTO 5
      0000                                <201>
51190 REM PRINT"SA=";SA;" N=";N
                   :REM UNNOETIG !!       <216>
51200 :GOTO 51060:REM NAECHST. SUCH-VERS. <192>
51300 REM NAECHSTES WORT                  <027>
51310 ZA=ZB+1                             <254>
51320 IF MID$(BE$,ZA,1)=""THEN 50530      <128>
51330 IF MID$(BE$,ZA,1)=" "THEN ZA=ZA+1:GO
      TO 51330                            <207>
51350 GOTO 50530                          <131>
51500 REM ICH KENNE SU$ NICHT ------------
      ------------------------------------
      --                                  <232>
51501 :                                   <168>
51510 I=INT(3*RND(1))+1                   <080>
51520 ON I GOTO 51530,51540,51550         <173>
51530 X$=""+"ICH KENNE DAS WORT '"+SU$+"'
      NICHT.":GOTO 52000                  <011>
51540 X$=""+"DAS WORT '"+SU$+"' IST MIR NI
      CHT BEKANNT.":GOTO 52000            <122>
51550 X$=""+"DEUTSCH IST LEIDER NUR MEINE
      ZWEITE SPRACHE. ICH KENNE DAS "     <167>
51555 X$=X$+"WORT '"+SU$+"' NICHT.":GOTO 5
      2000                                <158>
52000 REM FORMATIERTE TEXTAUSGABE --------
      ------------------------------------
      --                                  <120>
52010 IF X$=""THEN 52090                  <213>
52020 IF LEN(X$)<40 THEN PRINT X$:GOTO 520
      90                                  <248>
52030 IF LEN(X$)=40 THEN PRINT X$;:GOTO 52
      090                                 <148>
52040 I$=LEFT$(X$,41):IC=0:FOR I=1 TO LEN(
      I$):IF MID$(I$,I,1)=" "THEN IC=I    <242>
52050 NEXT I:IF IC=0 THEN PRINT X$:GOTO 52
      090                                 <064>
52060 I$=LEFT$(I$,IC-1):IF LEN(I$)=40 THEN
       PRINT I$;                          <046>
52070 IF LEN(I$)<40 THEN PRINT I$         <116>
52080 X$=RIGHT$(X$,LEN(X$)-IC):GOTO 52010 <206>
52090 RETURN                              <075>
52100 REM SATZ-TEILBEREICH LESEN ---------
      ------------------------------------
      --                                  <254>
52101 :                                   <004>
52110 HB%=SA/256:LB%=SA-HB%*256           <241>
52112 PRINT#15,"P";CHR$(3);CHR$(LB%);CHR$(
      HB%);CHR$(1)                        <170>
52114 INPUT#15,A:IF A=50 THEN PRINT"SATZNU
      MMER ZU HOCH"                       <245>
52120 IF A=50 THEN RETURN                 <165>
52130 INPUT#1,A$                          <199>
52140 WA=VAL(MID$(A$,22,1))               <030>
52150 WC=VAL(MID$(A$,23,3))               <089>
52160 A$=LEFT$(A$,21)                     <202>
52170 IF RIGHT$(A$,1)="{RIGHT}"THEN A$=LEF
      T$(A$,LEN(A$)-1):GOTO 52170         <154>
52180 RETURN                              <165>
53000 REM UP - TITELLINE -----------------
      ------------------------------------
      --                                  <170>
53001 :                                   <144>
53010 I1$="{37SPACE}":T$=""               <253>
53020 T$=T$+CHR$(18)+R$:I$=STR$(SC)+"/"+RI
      GHT$(STR$(MO),LEN(STR$(MO))-1)      <159>
53030 T$=T$+RIGHT$(I1$,41-LEN(T$)-LEN(I$))
      +I$+CHR$(146):PRINT"{HOME}";T$;     <240>
53040 PRINT"{22DOWN}";:RETURN             <073>
53200 REM UP - DISK EINLEGEN -------------
      ------------------------------------
      --                                  <023>
53201 :                                   <088>
53205 IF IC=21 THEN GOSUB 53700:REM WEITER <136>
53210 PRINT"{RVSON}BITTE DISK-SEITE";SIDE;
      "{LEFT,SPACE}EINLEGEN !{RVOFF}";:POK
      E 198,0:WAIT 198,1                  <119>
53220 PRINT"{29LEFT,29SPACE}";            <121>
53221 PRINT"{29LEFT}";                    <122>
53222 OPEN 2,8,2,"#":PRINT#15,"U1 2 0 18 0
      ":PRINT#15,"B-P";2;162:GET#2,I$     <037>
53224 CLOSE 2:IF SIDE<>VAL(I$)THEN 53200  <046>
53225 IF SIDE=2 THEN OPEN 1,8,3,"TEXTE"   <103>
53230 RETURN                              <199>
53400 REM ZEIGER POSTIONIEREN ------------
      ------------------------------------
      --                                  <108>
53401 :                                   <034>
53420 HB%=SA/256:LB%=SA-HB%*256:PRINT#15,"
      P";CHR$(3);CHR$(LB%);CHR$(HB%);CHR$(
      1)                                  <027>
53430 INPUT#15,IA:IF IA=50 THEN PRINT"SATZ
      NUMMER ZU HOCH"                     <222>
53440 RETURN                              <155>
```

**Listing 1. Ein »Adventure-Programmier-Gerüst« (Fortsetzung)**

```
53500 REM SATZ EINLESEN -----------------
      -----------------------------------
      --                                  <223>
53501 :                                   <134>
53510 IF A=50 THEN 53520                  <148>
53512 INPUT#1,I$                          <058>
53515 ZW=ZW+1:IF ZW>23 THEN GOSUB 53700:ZW
      =0                                  <010>
53520 RETURN                              <237>
53600 REM TEXT-EINLESEN -----------------
      -----------------------------------
      --                                  <121>
53601 :                                   <236>
53610 IF SIDE<>2 THEN SIDE=2:GOSUB 53200  <080>
53620 GOSUB 53400:REM POSITONIEREN        <165>
53630 GOSUB 53500:REM I$ EINLESEN         <068>
53640 IF RIGHT$(I$,1)<>"*"THEN 53660      <056>
53650 :IF I$<>"*"THEN PRINT LEFT$(I$,LEN(I
      $)-1)                               <118>
53652 :SA=0:RETURN                        <218>
53660 IF LEN(I$)<40 THEN PRINT I$         <182>
53670 IF LEN(I$)>=40 THEN PRINT I$;       <144>
53680 SA=SA+1                             <021>
53695 GOTO 53630                          <014>
53700 REM WEITER ------------------------
      -----------------------------------
      --                                  <152>
53701 :                                   <080>
53705 PRINT"[WEITER]";:POKE 198,0:WAIT 198
      ,1:PRINT"{8LEFT,8SPACE,4LEFT,UP}    <183>
53710 RETURN                              <171>
53800 REM KOMMENTAR LESEN ---------------
      -----------------------------------
      --                                  <213>
53801 :                                   <182>
53805 ZW=1                                <070>
53810 PRINT:GOSUB 53600:RETURN            <095>
53900 REM TEXTFORMER ------------------   <007>
53901 :                                   <026>
53902 IF X$<>""THEN PRINT                 <047>
53905 IF X$=""THEN 53945                  <230>
53910 IF LEN(X$)<40 THEN PRINT X$:GOTO 539
      45                                  <004>
53915 IF LEN(X$)=40 THEN PRINT X$;:GOTO 53
      945                                 <055>
53920 I$=LEFT$(X$,41):IC=0:FOR I=1 TO LEN(
      I$):IF MID$(I$,I,1)=" "THEN IC=I    <088>
53925 NEXT I:IF IC=0 THEN PRINT X$:GOTO 53
      945                                 <239>
53930 I$=LEFT$(I$,IC-1):IF LEN(I$)=40 THEN
       PRINT I$;                          <138>
53935 IF LEN(I$)<40 THEN PRINT I$         <203>
53940 X$=RIGHT$(X$,LEN(X$)-IC):GOTO 53905 <237>
53945 RETURN                              <152>
53950 REM ARTIKEL-WANDLER -------------   <116>
53951 :                                   <076>
53952 II$=LEFT$(OB$,3)                    <206>
53955 IF II$<>"DER"AND II$<>"DAS"THEN 5396
      5                                   <133>
53960 X$="":X$="HIER LIEGT EIN"+RIGHT$(OB$
      ,LEN(OB$)-3)+"."                    <095>
53965 IF II$<>"DIE"THEN 53970             <035>
53966 X$="":X$="HIER LIEGT EINE"+RIGHT$(OB
      $(I),LEN(OB$)-3)+"."                <045>
53970 GOTO 53905                          <070>
54000 REM RAUMTEXT-EINLESEN -------------
      -----------------------------------
      --                                  <055>
54001 :                                   <126>
54020 PRINT:FOR I=1 TO OZ :REM OB.TAB.    <187>
54030 :IF OB(I)<>ZN THEN 54080            <170>
54035 :IF OX(I)<1 THEN 54080  :REM NAMELOS <186>
54040 :SA=OX(I):GOSUB 53400 :REM POSIT.   <071>
54050 :IF OA(I)=1 THEN GOSUB 53500:R$=I$
             :REM READ I$, TITELLINE      <035>
54060 :IF WI(ZN)=1 AND VE<>11 AND OA(I)=1
      THEN PRINT R$:SA=0:GOTO 54080       <062>
54070 :GOSUB 53600          :REM LESEN    <009>
54080 NEXT I                              <059>
54081 FOR I=1 TO OZ                       <179>
54083 IF OB(I)<>ZN OR OA(I)<>3 OR OX(I)<>0
       THEN 54088                         <252>
54084 SA=OS(I):GOSUB 53400:GOSUB 53500:OB$
      ="D"+RIGHT$(I$,LEN(I$)-1)           <250>
54085 X$="HIER LIEGT "+OB$+".":ZW=ZW+1:GOS
      UB 53950                            <045>
54088 NEXT I                              <067>
54095 WI(ZN)=1:PRINT:RETURN               <205>
60000 REM TABELLEN EINLESEN -------------
      -----------------------------------
      --                                  <092>
60001 :                                   <030>
63000 REM ****************************    <017>
63005 REM * MINI-TEXTEDITOR V.III.4    *  <071>
63010 REM *       <C> 1985             *  <190>
63015 REM * VON MICHAEL NICKLES        *  <230>
63020 REM ****************************    <037>
63025 REM ANFANGSMENUE ----------------   <066>
63030 POKE 198,0:INPUT"{2DOWN}DATEINAME{2S
      PACE}*{3LEFT}";D$:POKE 198,0        <094>
63035 INPUT"SATZLAENGE{2SPACE}40{4LEFT}";S
      L:IF SL<1 OR SL>254 THEN PRINT"GEHT
      NICHT !":GOTO 30                    <211>
63040 OPEN 15,8,15                        <203>
63045 PRINT#15,"I0":OPEN 1,8,3,D$:PRINT   <200>
63050 REM ZWISCHENMENUE ---------------   <219>
63055 PRINT"{RVSON}S / L / ENDE{28SPACE,RV
      OFF}";                              <014>
63060 GET I$:IF I$=""THEN 63060           <252>
63065 IF I$="E"THEN PRINT"{CLR}";:GOTO 632
      65                                  <043>
63070 IF I$<>"S"AND I$<>"L"THEN 63060     <137>
63075 INPUT"SATZNR";SA:IF SA<0 OR SA>63000
       THEN 63070                         <050>
63080 IF I$="L"THEN PRINT"{CLR}";:GOTO 630
      95                                  <222>
63085 IF I$="S"THEN PRINT"{CLR}";:MO=1:GOT
      O 63150                             <126>
63090 GOTO 63050                          <212>
63095 REM LESEN -----------------------   <046>
63100 :PRINT"{CLR,RVSON}";SA;"{LEFT}-";STR
      $(SA+20);"{SPACE,RVOFF}"            <030>
63105 IA=SA                               <130>
63110 FOR SA=IA TO IA+20                  <040>
63115 : GOSUB 63195:GOSUB 63220           <042>
63120 : IF A=50 THEN SA=IA+21:GOTO 63135  <049>
63125 : IF LEN(A$)<40 THEN PRINT A$       <089>
63130 : IF LEN(A$)>=40 THEN PRINT A$;     <252>
63135 NEXT:PRINT                          <000>
63140 IF A=50 THEN PRINT"SATZNUMMER ZU HOC
      H!"                                 <120>
63145 GOTO 63050                          <011>
63150 REM SCHREIBEN -------------------   <138>
63155 PRINT"EINGABESATZ{2SPACE}//=KEINE EI
      NGABE"                              <185>
63160 GOSUB 63195:REM POSITIONIEREN       <146>
63165 IF A=50 THEN PRINT"SATZNUMMER ZU HOC
      H!":GOTO 63050:REM FEHLERKANALABFR. <009>
63170 GOSUB 63330:A$=BE$
                      :REM INPUT-ROUT.    <208>
63175 IF A$="//"THEN MO=0:GOTO 63050:REM E
      NDE                                 <157>
63180 GOSUB 63240:REM A$ SCHREIBEN        <131>
63185 SA=SA+1       :REM NAECHSTER SATZ   <149>
63190 GOTO 63160:REM NEXT (LOOP)          <010>
63195 REM POSITIONIEREN ***************   <220>
63200 HB%=SA/256:LB%=SA-HB%*256           <153>
63205 PRINT#15,"P";CHR$(3);CHR$(LB%);CHR$(
      HB%);CHR$(1)                        <085>
63210 INPUT#15,A                          <154>
63215 RETURN                              <022>
63220 REM SATZ EINLESEN ***************   <057>
63225 IF A=50 THEN RETURN                 <092>
63230 INPUT#1,A$                          <121>
63235 RETURN                              <044>
63240 REM SATZ SCHREIBEN **************   <043>
63245 IF A=50 THEN RETURN                 <114>
63250 IF A$=""THEN A$=":":REM ZUR SICHERHE
      IT !!!                              <141>
63255 PRINT#1,A$                          <228>
63260 RETURN                              <069>
63265 REM BEENDEN *********************   <107>
63270 CLOSE 1                             <032>
63275 CLOSE 15                            <120>
```

```
63280 PRINT "++ ENDE ++"                          <026>
63285 END                                         <038>
63290 REM KORREKTUR-ROUTINE ------------          <069>
63295 FOR I=0 TO LEN(BE$):PRINT CHR$(20);:
      NEXT I:BE$=""                               <123>
63300 SA=SA+SC          : REM TEXTZEIGER          <015>
63305 IF SA<0 THEN SA=0 : REM SPERRE              <117>
63310 GOSUB 63195: REM POSITIONIEREN              <042>
63315 IF A=50 THEN SA=SA-SC:GOTO 63310            <132>
63320 INPUT#1,A$                                  <213>
63325 RETURN                                      <134>
63330 REM TEXTEINGABE ------------------          <248>
63335 IF AR=1 THEN AR=0:A$="{RVSON}SATZNUM
      MER ZU HOCH!{RVOFF}":GOTO 63345             <104>
63340 GOSUB 63195:GOSUB 63220                     <190>
63345 PRINT"{HOME,40SPACE}";:PRINT"{HOME,R
      VSON}SATZ";SA;"{RVOFF}"                     <042>
63350 PRINT"{40SPACE,UP}";A$                      <203>
63355 PRINT"TTTTTTTTTTTTTTTTTTTTTTTTTTTTTT
      TTTTTTTTTT";                                <086>
63360 PRINT"{DOWN}"                               <059>
63365 BE$="":POKE 198,0:POKE 211,0:POKE 21
      4,22:SYS 58732:PRINT"{YELLOW}@";            <182>
63370 GET X$:IF PEEK(203)=1 THEN 63430            <129>
63375 IF PEEK(203)=7 AND PEEK(653)=0 THEN
      SC=1 : GOSUB 63290: GOTO 63330              <025>
63380 IF PEEK(203)=7 AND PEEK(653)=1 THEN
      SC=-1 : GOSUB 63290: GOTO 63330             <162>
63385 IF X$=""THEN 63370                          <154>
63390 IF LEN(BE$)=0 AND ASC(X$)=20 THEN 63
      370                                         <125>
63395 I=ASC(X$):IF I<32 OR I>133 AND I<159
       THEN IF I<>20 THEN 63370                   <155>
63400 IF LEN(BE$)=SL AND I<>20 THEN 63370         <147>
63405 IF X$=" "THEN X$="{RIGHT}"                  <134>
63410 BE$=BE$+X$                                  <153>
63415 PRINT CHR$(20);X$;"@";                      <225>
63420 IF I=20 THEN BE$=LEFT$(BE$,LEN(BE$)-
      2):GOTO 63370                               <058>
63425 GOTO 63370                                  <005>
63430 PRINT"{LEFT,SPACE}":GOSUB 63195             <025>
63435 IF BE$=""THEN BE$=":"                       <146>
63440 PRINT"{HOME,24DOWN}":RETURN                 <122>
```

**Listing 1. Ein »Adventure-Programmier-Gerüst« (Schluß)**

## Dokumentation

3-4 Laden und Aktivieren des deutschen Zeichensatzes (muß sich ebenfalls auf der ersten Diskseite befinden!).

5 Floppykanal öffnen.

6 Disk Seite 2 einlegen!

15 Ab 60000 stehen alle Tabellen wie der Wortschatz-Index.

20 Anhand des Feldes WI(1)-WI(100) merkt sich das Programm, in welchem Raum der Spieler bereits war. Genau wie bei INFOCOM erhält der Spieler von Räumen, in denen er bereits war, nur noch eine kurze Beschreibung.

30 Zähler für Punktzahl und Anzahl der Befehlseingaben werden auf Null gesetzt.

40 Der Titelvorspann des Spiels wird von der relativen Datei aufgerufen (SA ist dann meistens 1, da der Vorspann meistens beim ersten Satz der Texte-Datei beginnt).

50 Raumbeschreibung des ersten Zimmers ausgeben.

50000-52180 Hier steht Parser 4.0. Allerdings sind einige kleine Sachen hinzugefügt. Kommt im Befehlssatz das Wort »Alles« vor, so wird der Variablen SP der Wert 1 zugeordnet. Kommt »Ausser« vor, so erhält die Variable SQ den Wert 1. Dies ist nötig, damit wir auch Sätze wie zum Beispiel »NIMM ALLES AUSSER DEM SCHWERT« abhandeln können.

53000-53040 Unterprogramm zum Einblenden einer

Statuszeile in der obersten Bildschirmzeile. Wie bei Infocom stehen in dieser Zeile der Name des Raumes, in dem man sich gerade befindet, sowie Punktzahl und Anzahl der Versuche. Dieses Unterprogramm wird innerhalb der erweiterten INPUT-Routine von Parser 4.0 aufgerufen (Zeile 50011).

53200–53230 Dieses Unterprogramm haben wir bereits im Kapitel Dateiverwaltung kennengelernt. Es dient dazu, festzustellen, ob die richtige Diskseite (Wert von Variable SI) eingelegt ist. Ist Disk Seite 2 eingelegt, so öffnet das Unterprogramm außerdem automatisch die relative Datei »Texte«.
Soll Disk Seite 1 eingelegt werden, so sagen Sie einfache SIDE=1:GOSUB53200. Der Rücksprung aus dem Unterprogramm erfolgt erst dann, wenn die gewünschte Seite eingelegt ist.

53400–53440 Das altbekannte Unterprogramm zum Positionieren des Zeigers auf eine relative Datei.

53500–53520 Dito, zum Einlesen eines Satzes (I$) aus einer relativen Datei.

53600–53695 Ein Text aus der relativen Texte-Datei wird eingelesen. Beginnend ab Satz SA bis zu dem nächsten Satz, der mit einem Sternchen endet. Wenn Sie also einen Text von der relativen Datei einlesen und auf dem Bildschirm ausgeben wollen, so ist ein
`SA = (1. Satz des Textes): GOSUB 53600` notwendig.

53700–53710 Auf dem Bildschirm erscheint das Wort WEITER. Bei Tastendruck verschwindet es wieder, und das Unterprogramm wird mittels RETURN beendet. Dieses Unterprogramm wird zum Beispiel dann verwendet, wenn ein Text am Stück auf dem Bildschirm ausgegeben wird, der mehr als 25 Sätze enthält. Nach dem 24. Satz erscheint WEITER, damit dem Spieler genug Zeit zum Lesen bleibt und der Text nicht weg-scrollt.



53800–53810 Dasselbe wie 53600–53695 nur, daß von der eigentlichen Textausgabe diesmal eine Leerzeile eingefügt wird.

53900–53945 Der altbekannte Textformer, der den String X$ per word-wrapping auf dem Bildschirm ausgibt.

53950–53970 Die Gegenstände beziehungsweise transportablen Objekte sind in der Texte-Datei mit voranstehendem Geschlechtsartikel – der, die oder das – gekennzeichnet. Am Ende der Raumbeschreibungen soll jedoch für zum Beispiel Schwert der Kommentar »Hier liegt ein Schwert« kommen. (Natürlich wieder analog zu Infocoms THERE IS A SWORD LYING HERE.)
Die Artikelwandlung ist ganz einfach:
Aus der beziehungsweise das wird EIN. Aus die wird EINE.

54000–54095 Unterprogramm zum Einlesen eines Raumtextes. Das Einlesen eines Raumtextes ist nicht ganz so einfach wie das Einlesen eines gewöhnlichen Textes, da sich ein Raumtext aus mehreren Einzelteilen zusammensetzt:
– Zuerst wird der Haupt-Raumtext ausgegeben. Das ist der Text, der sich nie ändert.
– Dann werden Textteile ausgegeben, die sich ändern können. Ein Beispiel: Eine Tür, die bei der Beschreibung sowohl als offen als auch als geschlossen beschrieben werden kann.
– Im Raum befindliche Gegenstände. Dies sind in der Regel Objekte, die transportiert werden können. In der Raumbeschreibung erscheinen diese Objekte dann als »Hier liegt ein...« etc.

63000–63440 Hier steht eine Mini-Version des Relativen-Dateien-Editors. Er wird mittels GOSUB 63000 gestartet. Beim Schreiben eines Textadventures fallen einem stets neue Kommentare ein, die man der Texte-Datei zufügen muß. Es wäre viel zu umständlich, wegen jedem neuen Kommentar die aktuelle Arbeit zu speichern, dann den Relative-Datei-Editor laden etc. Daher steht am Ende des Gerüstes eine Mini-Version, mit der nur gelesen und geschrieben werden kann. Am Ende der Programmierung kann dieser Editor entfernt werden.

Die Dokumentation ist an dieser Stelle noch nicht komplett. Sie müssen erst noch einige andere Dinge wissen.

## Spielablauf der ersten Szene

So läuft die erste Spielszene ab: Der Spieler erwacht in der Glaskuppel. Er muß jetzt im Sand graben, um einen Zettel und ein Amulett zu finden. Anschließend gelangt er durch den Tunnel in eine Kristallgrotte, in der eine Statue steht. Im Marmorboden steckt ein Schwert, das sich nicht herausziehen läßt. Der Spieler muß jetzt die Statue berühren, um sich in die Figur zu verwandeln, die die Statue darstellt. Er nimmt so den Körper eines jungen Kriegers an. Jetzt kann er mühelos das Schwert aus dem Boden ziehen. Dieses Herausziehen des Schwertes wirkt gleichzeitig wie ein Schlüssel: In der westlichen Wand tut sich eine Öffnung auf, die zu einem Felsplateau hinausführt. Dort beginnt die nächste Szene.

Wir wollen jetzt gemeinsam das Gerüst um diese Szene ergänzen. Sie erfahren dabei gleichzeitig die Dokumentation zu den übrigen Gerüstteilen.

```
60000 REM TABELLEN EINLESEN --------------
      -----------------------------------
      --                                  <092>
60001 :                                   <030>
60100 REM WORTSCHATZ-INDEX -----------    <213>
60110 DIM IN(26)                          <192>
60120 FOR I=0 TO 26:READ IN(I):NEXT I     <181>
60130 DATA 1,4,0,29,32,48,53,64,72,0,74,82
      ,0,88,96,100,107,109,124,160,167    <214>
60140 DATA 170,190,0,0,201                <188>
60150 DATA 211 : REM BUCHSTABE NACH Z !   <122>
60200 REM OBJEKT-TABELLE ------------     <087>
60210 OZ=13:REM ANZAHL DER OBJEKTE        <133>
60220 DIM OB(OZ),OX(OZ),OA(OZ),OU(OZ),OS(O
      Z)                                  <103>
60230 FOR I=1 TO OZ                       <232>
60240 : READ OB(I),OX(I),OA(I),OU(I),OS(I) <003>
60250 NEXT I                              <133>
60260 DATA 1,24 ,1,24 ,164:REM GLASKUPPEL <183>
60262 DATA 1,0  ,2,95 ,165:REM SAND       <198>
60264 DATA 2,34 ,1,34 ,166:REM TUNNEL     <039>
60266 DATA 3,43 ,1,43 ,167:REM KR.GROTTE  <066>
60268 DATA 3,102,2,0  ,168:REM QUADER     <167>
60270 DATA 3,0  ,2,113,169:REM STATUE     <177>
60272 DATA 3,106,0,0  ,170:REM SCHWERT    <131>
60274 DATA 0,0  ,3,79 ,171:REM ZETTEL     <127>
60276 DATA 0,0  ,3,87 ,172:REM AMULETT    <092>
60278 DATA 0,76 ,2,91 ,173:REM GRUBE      <098>
60279 DATA 0,161,2,0  ,174:REM OEFFNUNG   <127>
60280 DATA 4,54 ,1,54 ,175:REM FELSPLAT.  <123>
60282 DATA 3,0  ,2,97 ,176:REM KRISTALLE  <037>
60500 REM LANDKARTE ----------------      <047>
60501 :                                   <022>
60510 DIM MAP(4,10)                       <148>
60520 FOR I=1 TO 4                        <037>
60530 : FOR I1=1 TO 10                    <003>
60540 :    READ MAP(I,I1)                 <126>
60550 : NEXT I1                           <021>
60560 NEXT I                              <189>
60570 DATA 2,0,0,0,0,0,0,0,0,0            <042>
60572 DATA 3,1,0,0,0,0,0,0,0,0            <084>
60574 DATA 0,2,0,0,0,0,0,0,0,0            <094>
60576 DATA 0,0,0,3,0,0,0,0,0,0            <038>
60600 RETURN                              <203>
```

**Listing 2. Ergänzung zu Listing 1**

Zunächst müssen ab Zeile 60000 die Tabellen fertiggestellt werden, die im Rahmen von Gordon Saga II benötigt werden: die Wortschatz-Index-Tabelle, die Landkarte und die Objekt-Logik-Tabelle. Bitte ergänzen Sie das bisherige Gerüst-Programm um die Zeilen aus Listing 2.

## Anlegen der Tabellen

Auf den Wortschatz-Index brauche ich nicht lange einzugehen. Es ist der Index des modifizierten Wortschatzes – erstellt mit dem Index-Maker-Programm und anschließend in Data-Zeilen abgelegt. Anders bei der Objekttabelle. Die Objekttabelle ist das wichtigste am ganzen Programm. Zu jedem der Objekte sind mehrere Daten gespeichert. Diese Daten sind in den Feldern OB(), OX(), OA(), OU() und OS() untergebracht. Was geben diese einzelnen Felder an? Betrachten wir sie am besten einmal im einzelnen:

In der Variablen OZ ist die Anzahl der Objekte gespeichert.

OB(1)–OB(Z): OB(X) gibt an, in welchem Raum sich Objekt X befindet. Der Wert OB(X) ist immer dann größer Null, wenn sich das Objekt X in einem Raum befindet.
Ist OB(X) = 0, so heißt dies, daß das Objekt X nicht bzw. nicht mehr vorhanden ist.
Ist OB(X) kleiner als Null, so befindet sich das Objekt irgendwo anders als in einem Zimmer. Zum Beispiel im Besitz des Spielers –OB(X)=–1. Soll das Objekt in einem Schrank liegen, so kann man ihm eine für den Schrank spezifische negative Zahl zuordnen und so weiter.

OX(1)–OX(OZ): OX(X) gibt an, ab welchem Satz der relativen Datei der Beschreibungs-Text zu Objekt X steht. Ist OX(X)=0, so heißt dies, daß das Objekt X bei der Raumbeschreibung nicht anhand eines Textes erläutert wird.

OA(1)–OA(OZ): OA(OZ) gibt an, um welche Art von Objekt es sich bei Objekt X handelt. Es gibt folgende Werte:
1. Das Objekt ist ein Raum (zum Beispiel Tunnel).
2. Das Objekt befindet sich im Raum und kann nicht transportiert werden (zum Beispiel Öffnung in der Wand).
3. Das Objekt kann transportiert werden vom Spieler (zum Beispiel Schwert).

OU(1)–OU(OZ): OU(OZ) zeigt auf den ersten Satz des Kommentars in der Texte-Datei, der ausgegeben wird, wenn das Objekt untersucht wird. Wird ein Objekt untersucht, dessen OU(X)-Wert gleich 0 ist, so erfolgt der Kommentar »DAS OBJEKT X HAT KEINE BESONDEREN MERKMALE.«

OS(1)–OS(OZ): OS(X) gibt an, in welchem Satz der Texte-Datei der Objektname von Objekt X gespeichert ist (Beispiel: die Glaskuppel = 164).

Die Objekttabelle steht nur bei der Entwicklung des Programmes in DATA-Zeilen. Sie wird später entweder als SEQ-File geladen oder als PRG-File geladen und irgendwo im RAM mit entsprechender Verwaltung untergebracht. Durch geschicktes Manipulieren der Objekttabelle ist es leicht, dem Adventure Intelligenz zu verleihen: Das Programm lacht den Spieler aus, wenn er auf die unsinnige Idee kommt, einen Raum wie den Tunnel nehmen zu wollen. Will der Spieler eine lebende Person nehmen (OA-Wert könnte zum Beispiel mit 4 definiert werden), so kann sich der Analyseteil des Adventures mit »ICH GLAUBE KAUM, DASS SICH DIE PERSON X SO OHNE WEITERES NEHMEN LÄSST« etc.

Nehmen wir einmal an, der Spieler will Objekt X nehmen – also BES=»NIMM OBJEKT X«.

Das Programm kann nun verschiedenes überprüfen und so feststellen, ob der Spieler Objekt X überhaupt nehmen kann.

Erweist sich OB(X) als –1, so meldet sich das Adventure mit »SIE HABEN OBJEKT X BEREITS«.

Kristallgrotte 0/5 } I
Kristallgrotte
Dies ist eine riesige Grotte, deren Wände mit Kristallen, die in allen Farben des Lichtspektrums glitzern, besetzt sind. Der Boden besteht aus glatten, strahlend weißen Marmorplatten. Trotz zahlreichen Unebenheiten ähnelt diese Grotte dem Inneren einer Halbkugel mit einem Radius von ca. 100 Metern. Der einzige Ausgang von hier ist ein in südliche Richtung verlaufender Tunnel. } II
Auf dem Marmorboden, exakt im Zentrum des Raumes, steht ein goldener Quader, von dem eine ca. drei Meter hohe Statue emporragt. Nur wenige Meter von der Statue entfernt steckt ein goldenes Schwert zu etwa einem Drittel seiner Gesamtlänge im Marmorboden. } III
Hier liegt ein Zettel.
Hier liegt ein Amulett. } IV

**Bild 2. Aufteilung der Raumbeschreibung**

IST OB(X) ungleich ZN (der aktuelle Raum), so erfolgt der Kommentar »OBJEKT X IST HIER NICHT ZU SEHEN«. Außerdem läßt das Programm nur zu, Objekte zu nehmen, deren OA-Wert = 3 ist (also transportables Objekt beziehungsweise Gegenstand).



Mit den 5 Objekttabellen kann also jedes Objekt exakt analysiert werden. Ein Beispiel: Die Statue hat folgende Werte:

OB(6)=3: Das heißt, die Statue steht im Raum 3.

OX(6)=0: Das heißt, die Statue wird in der Raumbeschreibung nicht mehr besonders erwähnt. Sie kommt schließlich bereits in der Beschreibung des Quaders, auf dem sie steht, vor.

OA(6)=2: Das heißt, die Statue ist ein feststehendes Objekt und kann daher auch nicht transportiert werden.

OU(6)=113: Ab Satz 113 steht der Kommentar für die Untersuchung der Statue.

OS(6)=164: In diesem Satz der Texte-Datei steht der Objektname der Statue – die Statue.

Ganz klar wird Ihnen die Objekttabelle allerdings erst dann werden, wenn Sie sich mit der Abhandlung der Objekte im Programm selbst vertraut gemacht haben.

Zunächst möchte ich Ihnen jedoch noch einmal ganz genau zeigen, wie sich eine Raumbeschreibung zusammensetzt. Bild 2 zeigt die Raumbeschreibung der Kristallgrotte, nachdem man dort noch zwei transportable Gegenstände verloren hat.

I Dies ist die Statuszeile. Sie ist ständig zu sehen und gibt an, in welchem Raum man sich gerade befindet, die bisher erzielte Punktzahl und die Anzahl der Spielzüge, die man bisher gemacht hat.

II In diesem Bereich steht die Hauptraumbeschreibung – also die Beschreibung, die sich während des gesamten Verlaufs des Spieles nicht ändert.

III Hier sind alle Objekte aufgeführt, die sich zwar verändern können, jedoch (zumindest im Moment) nicht vom Spieler transportiert werden können. Der OA-Wert dieser Objekte beziehungsweise Beschreibungen ist 2. Der OA-Wert von Gegenständen (=transportable Objekte) ist in der Regel 0. Hier gibt es jedoch eine Ausnahme: Es ist möglich, ein solches Objekt bei sei-

nem ersten Auftreten ausführlicher zu beschreiben als mit HIER LIEGT EIN ... Wird das Objekt jedoch genommen, so wird innerhalb der NIMM-Routine sein OA-Wert für immer auf Null gesetzt. Wird es wieder verloren, so taucht es in der Raumbeschreibung anschließend mit der Beschreibung HIER LIEGT EIN ... auf.

IV Ganz am Ende der Raumbeschreibung werden die Objekte aufgeführt, die dort vom Spieler verloren worden sind beziehungsweise bei denen es sich um Gegenstände handelt, deren OA-Wert von vorneherein 0 ist.

Das Unterprogramm zur Erzeugung der Raumbeschreibung steht im Programm ab Zeile 54000. Vergleichen Sie doch mal die obige Theorie mit diesem Programmteil – dann gewinnen Sie am schnellsten den Durchblick!

Die nächste Tabelle, die definiert wird, ist die Landkarte, die als Feld MAP(N,10) angelegt ist, wobei N für die Anzahl der Räume steht. In dieser Tabelle ist eingetragen, welche Verbindungen zwischen den einzelnen Räumen existieren. Dabei werden allerdings nur die stets bestehenden Verbindungen eingetragen, die nicht erst geschaffen werden müssen (zum Beispiel durch das Öffnen einer Tür mit einem Schlüssel). Die Richtungen stehen so: N,S,W,O,NW, NO,SW,SO, rauf, runter – also MAP(N,1) für wohin es nach Norden geht und MAP(N,10) für Süden. Eine nicht existente Verbindung hat den Wert 0.

Beispiel: Ist MAP(3,3)=4, so heißt dies, daß man von Raum 3 nach Westen gehen kann, um zu Raum 4 zu gelangen.

## Analyseteil

Wie schon beim ersten Adventure-Kurs unterscheiden wir auch diesmal zwei Arten von Action: allgemeine und raumspezifische Action.

Allgemeine Action sind Handlungen, die man in jedem Raum, jederzeit ausführen kann – zum Beispiel Objekte untersuchen, Sachen nehmen, verlieren, Inventur etc.

Raumspezifische Action hingegen sind Handlungen, die man nur in bestimmten Räumen ausführen kann. »Berühre Statue« hat nur in dem Raum einen Effekt, in dem die Statue steht.

Im Programm steht die allgemeine Action in den Zeilen 300–800, die raumspezifische Action ab Zeile 2000.

Beim Gerüst sind in den Zeilen 300 – 700 bisher nur die Routinen untergebracht, die bei jedem Adventure unentbehrlich sind: GEH-Routine, NIMM-Routine, INVENTUR-Routine, VERLIER-Routine, UNTERSUCHE-Routine.

Ich verzichte an dieser Stelle bewußt auf eine genaue Dokumentation dieser Programmteile: Die Routinen sind sauber programmiert, und es sollte Ihnen keine Schwierigkeiten bereiten, sie selbst zu analysieren, wenn Sie bisher aufmerksam mitgearbeitet haben!

Wenn Sie das bisherige Programm um die folgenden wenigen Zeilen ergänzen, haben Sie die erste Szene von Gordon Saga Teil II komplett im Computer, da diese Zeilen die noch fehlende raumspezifische Action darstellen.

800–900 Hier ist bei der allgemeinen Action noch eine kleine Routine eingefügt, mit der man den Brief lesen kann, der in Gordon Saga Teil 2 gefunden wird. Diese Routine habe ich erst jetzt beigefügt, da sie zwar zur allgemeinen Action zählt (man kann einen Brief überall lesen – nicht raumabhängige Action), aber nicht für jedes Adventure benötigt wird.

2100–2130 Hier steht die raumspezifische Action zur Glaskuppel.

2110 Hier wird festgestellt, ob der Spieler im Sand graben will: ist Verb = 15 (graben) und Objekt = 2 (Sand)?

```
800 REM LESEN --------------------          <182>
805 IF VE<>16 THEN 900                       <198>
810 IF O1=8 AND OB(8)=-1 THEN SB=80          <056>
900 :                                        <114>
2000 REM RAUMSPEZIFISCHE ACTION ----------
     -------------------------------------
     -                                       <056>
2010 ON ZN GOTO 2100,2200,2300,2400          <139>
2100 REM GLASKUPPEL                          <069>
2110 IF VE<>15 OR O1<>2 THEN 2120            <005>
2115 IF PA(0)<3 THEN SB=71:PA(0)=PA(0)+1     <114>
2117 IF PA(0)=2 THEN SB=73:OB(8)=ZN:OB(9)=
     ZN:OB(10)=1                             <035>
2119 IF PA(0)=3 THEN SB=72                   <112>
2120 IF VE=1 AND O1=3 THEN RI=1:GOTO 300     <245>
2130 GOTO 100                                <026>
2200 REM TUNNEL                              <039>
2210 GOTO 100                                <106>
2300 REM KRISTALLGROTTE                      <198>
2305 IF VE=17 AND O1=6 AND OU(6)=113 THEN
     SB=119:OU(6)=135:GOTO 2307              <081>
2306 IF VE=17 AND O1=6 AND OU(6)=135 THEN
     PRINT"ES PASSIERT NICHTS."              <032>
2307 IF VE<>12 OR O1<>7 THEN 2310            <010>
2308 IF OU(6)=113 THEN SB=110                <202>
2309 IF OU(6)=135 THEN SB=139:OA(7)=3:OB(7
     )=-1:OB(11)=3:MAP(3,3)=4:OX(7)=0        <152>
2310 IF VE=1 AND O1=3 THEN RI=3:GOTO 300     <183>
2320 GOTO 100                                <218>
2400 REM FELSPLATEAU                         <194>
2410 GOTO 100                                <052>
```

**Listing 3. Damit ist Gordon Saga II komplett**

Gräbt der Spieler zum ersten Mal, so wird der Variablen SB der Wert 71 gesetzt (Zeiger auf Satz: Sie graben in dem Sand.).

In Zeile 110 wird der Wert der Variablen SB abgefragt. Ist SB ungleich Null, so wird in der Texte-Datei der Kommentar ausgelesen, der ab Satz SB bis zum nächsten Sternchen steht. SB wird sodann auf 0 gesetzt. Mittels SB werden also Kommentare aus der Texte-Datei abgerufen.

2117 Zweites Graben. Der Spieler findet ein Amulett und einen Zettel. Diesen beiden Objekten wird jetzt der OB-Wert der Zimmernummer ZN zugeordnet, da diese beiden Objekte gefunden sind und von nun an im Spiel enthalten sind.
Außerdem wird der OB-Wert der Grube auf ZN gesetzt. Dadurch wird die Grube das nächste Mal in der Raumbeschreibung erwähnt, wenn der Spieler Raum 1 untersucht: Es scheint so, als ob hier jemand vor kurzem gegraben hat.....Durch solche Action werden die Raumbeschreibungen variabel – das Adventure gewinnt an »Leben«.

2119 Versucht der Spieler ein drittes Mal zu graben, so erfolgt der Kommentar, daß weiteres Graben sinnlos ist.

2120 Wenn Verb=geh und Objekt=Tunnel, dann ist dies gleichbedeutend mit der Befehlseingabe »Norden«, da man von der Glaskuppel nach Norden gehen muß, um in den Tunnel zu gelangen. Also einfach die RI-Variable auf Norden setzen – RI=1 – und Sprung zur GEH-Routine (GOTO 300).

2200 Im Tunnel kommt keine raumspezifische Action vor.

2300–2320 Raumspezifische Action der Kristallgrotte.

2305 Wenn Verb=berühre und Objekt=Statue then Kommentar ab Zeile 119: Die Statue hat jetzt die Form des Spielers angenommen. Beim Untersuchen derselben muß also ein anderer Kommentar als bisher erscheinen. Also OU(6)=135.

Sie sehen schon: Mit Hilfe der einzelnen Objekt- und

Wortschatz-Tabellen läßt sich die raumspezifische Action nahezu in Klartext übersetzen.

Damit wäre ich mit meinen Erklärungen auch schon am Ende. Die restlichen Action-Szenen können Sie leicht selbst deuten. Adventure-Programmierung ist ein sehr kompliziertes Gebiet. Ich bin mir sicher, daß einige von Ihnen bereits jetzt Ideen haben, wo sich hier und da noch so manches verbessern läßt.

Vielleicht haben Sie auch schon die Idee zu Parser 6.0 – dem NON-PLUS-ULTRA-Parser im Kopf.

Ich hoffe, Ihnen mit diesem Kapitel neue Anregungen im Bereich der Adventure-Programmierung gegeben zu haben. Wenn Sie noch irgendwelche Fragen oder Probleme mit Adventure-Programmierung haben, dann wenden Sie sich doch mal an die 64'er, vielleicht geben Ihre Nachfragen Anlaß zu einem kleinen Folgekurs.

Ich will allerdings noch nicht zum Schlußwort kommen, denn es steht noch ein wichtiges Kapitel aus:

## Der Adventure-Cracker

Das Programm im Listing 4 macht's möglich: Adventures im Schlaf gelöst. Naja, ganz im Schlaf geht's nicht, aber Sie müssen sich lediglich im Sessel zurücklehnen und am Bildschirm mitverfolgen, wie der gesamte Text des Adventures ausgegeben wird.

Im Prinzip ist der Adventure-Cracker nichts weiter als ein ultra-primitiver Disk-Monitor, mit dem sich die einzelnen Bytes einer Diskette lesen lassen. Sie müssen einfach die Diskette, auf der sich das Adventure befindet, ins Laufwerk legen und dann mittels Angabe von Track und Sector nach Text suchen.

Der Adventure-Cracker filtert automatisch alle Zeichen heraus, die keinen Text darstellen (Grafikzeichen etc.). Es ist nur eine Frage der Zeit, bis man so den gesamten Text eines Adventures herausfindet.

Der Cracker knackt alle Basic-Adventures, Adventures, die mit relativen Dateien arbeiten (also unsere), und auch professionelle Spiele. Den INFOCOMs kann man mit dem Adventure-Cracker allerdings in keinster Weise zu Leibe rücken: Ihr Text ist auf eine äußerst komplizierte Weise verschlüsselt und bringt auch harte Dechiffrierer zur Verzweiflung ...

Es ist also ratsam, Adventure-Texte nicht einfach unverschlüsselt in relativen Dateien abzulegen. Parallel zum Codieren bietet sich allerdings noch etwas anderes an:

## Text-Raffung

Es gibt verschiedene Methoden, Text zu kürzen, um so letztendlich noch mehr Text auf einer Disk-Seite unterzubringen. Es bietet sich zum Beispiel an, einzelne Kürzel zu definieren. Schließlich haben wir 255 verschiedene Zeichen zur Verfügung, von denen zur reinen Textdarstellung nur ca. 70 benötigt werden (Groß-Klein-Buchstaben und diverse Satzzeichen). Man könnte nun doch eigentlich eine Tabelle anlegen, in der häufige Buchstabenkombinationen (er, die, ch, das, ng, en, .....) durch Grafikzeichen codiert werden.

Sie können insgesamt zirka 180 solche Kürzel definieren und dann ein Programm schreiben, das Ihre relativen Dateien automatisch kürzt und die Buchstaben anhand einer weiteren Tabelle – einem Code-Schlüssel – chiffriert.

Es gibt unglaublich viele Verschlüsselungs- und Text-Raffungsmethoden, die sich relativ einfach programmieren lassen. Denken Sie mal nach!

## Bonbon zum Schluß

Zu Beginn dieses Kurses habe ich Ihnen versprochen, daß das freie RAM bis zum letzten Byte mit interessanten Routinen belegt wird. Das haben wir auch geschafft. Bis auf einen kleinen Bereich: von 1024 bis 2023 der alte Bildschirmspeicher.

Dort bringen wir jetzt noch eine Maschinenroutine unter, die in Listing 5 abgedruckt ist.

Diese Routine können Sie erst dann laden, wenn der deutsche Zeichensatz aktiviert und der Bildschirm an eine andere Stelle als 1024–2023 gelegt worden ist. Zum Laden können

```
0 REM ********************************        <138>
1 REM *                              *        <050>
2 REM *    ADVENTURE-CRACKER         *        <187>
3 REM *       VERSION 1.0            *        <247>
4 REM *   VON MRIGHT$CHAEL NICKLES   *        <015>
5 REM *                              *        <054>
6 REM ********************************        <144>
10 PRINT"{CLR,RVSON}ADVENTURE-CRACKER VERS
   ION 1.0{11SPACE,RVOFF}";                   <038>
20 PRINT"{DOWN}BITTE DISK MIT SPIEL EINLEG
   EN{SPACE,RVSON}RETURN{RVOFF}"              <022>
25 POKE 198,0:WAIT 198,1                      <253>
30 OPEN 1,8,15:OPEN 2,8,2,"#"                 <067>
50 REM HAUPTMENUE                             <188>
51 PRINT"{RVSON}EINZEL / ALLES"               <099>
52 GET X$:IF X$=""THEN 52                     <225>
55 IF X$="E"THEN 200                          <166>
57 IF X$="A"THEN 100                          <162>
58 GOTO 52                                    <044>
100 REM ALLE DATEN VON DISK LESEN             <147>
105 INPUT"AB WELCHEM TRACK ";AN               <041>
110 FOR T=AN TO 35: REM TRACKS                <192>
112 REM AS=ANZAHL DER SEKTOREN                <153>
113 IF T<18         THEN AS=21                <216>
114 IF T>17 AND T<25 THEN AS=19               <197>
115 IF T>24 AND T<31 THEN AS=18               <008>
116 IF T>30         THEN AS=17                <081>
118 FOR S=1 TO AS: REM SECTORS                <063>
120 GOSUB 1000:REM DATEN IN BUFFER LESEN      <093>
130 GOSUB 2000:REM DATEN AUSDRUCKEN           <051>
140 NEXT S                                    <048>
150 NEXT T                                    <066>
200 REM EINZELNE DATEN LESEN                  <043>
201 PRINT                                     <047>
205 INPUT"TRACK ";T                           <145>
206 INPUT"SEKTOR ";S                          <087>
210 IF T<1 OR T>35 THEN 200                   <182>
211 IF S<1 THEN 200                           <142>
212 IF T<18 AND S>21 THEN 200                 <213>
213 IF T>17 AND T<25 AND S>19 THEN 200        <242>
214 IF T>24 AND T<31 AND S>18 THEN 200        <054>
215 IF T>30 AND S>17 THEN 200                 <034>
220 GOSUB 1000                                <176>
230 GOSUB 2000                                <194>
250 GOTO 200                                  <186>
1000 :                                        <214>
1001 REM DATEN IN BUFFER LESEN                <212>
1002 :                                        <216>
1005 PRINT#1,"U1 2 0 ";T;S                    <068>
1010 RETURN                                   <050>
2000 :                                        <198>
2001 REM BUFFER AUSLESEN                      <114>
2002 :M$=""                                   <099>
2010 FOR I=0 TO 255                           <074>
2020 : GET#2,A$                               <046>
2021 IF A$=""THEN 2030                        <195>
2022 M=ASC(A$) : REM CHECKEN                  <008>
2023 IF M>64 AND M<91 THEN PRINT A$;          <189>
2024 IF M>96 AND M<123 THEN PRINT A$;         <004>
2030 IF M=46 OR M=58 OR M=59 OR M=63 OR M=
     32 OR M=160 OR VAL(A$)>0 THEN PRINT A
     $;                                       <155>
2040 NEXT I                                   <090>
2050 RETURN                                   <076>
```

Listing 4. Der Adventure-Cracker

```
programm : ms                    1400 16b0
------------------------------------------
1400 : a2 0f 20 c9 ff 20 fd ae   c1
1408 : 20 8a ad 20 f7 b7 a9 50   61
1410 : 20 d2 ff a9 08 20 d2 ff   9b
1418 : a5 14 20 d2 ff a5 15 20   eb
1420 : d2 ff a9 01 20 d2 ff 20   55
1428 : cc ff a2 08 20 c6 ff a2   1b
1430 : 00 20 cf ff 9d fa 04 c9   89
1438 : 0d f0 03 e8 d0 f3 e0 28   1c
1440 : d0 05 a9 00 9d fa 04 e8   91
1448 : a9 00 9d fa 04 20 cc ff   2c
1450 : a2 ff e8 bd fa 04 d0 fa   ed
1458 : ca bd fa 04 c9 0d f0 f8   fb
1460 : c9 2a f0 2f a2 00 bd fa   77
1468 : 04 f0 06 20 d2 ff e8 d0   dc
1470 : f5 a5 14 18 69 01 85 14   1d
1478 : a5 15 69 00 85 15 ee f9   b3
1480 : 04 ad f9 04 c9 16 d0 03   f0
1488 : 20 b9 04 a2 0f 20 c9 ff   f3
1490 : 4c 0e 04 a9 0d 9d fa 04   cb
1498 : e8 a9 00 9d fa 04 a2 00   63
14a0 : bd fa 04 f0 06 20 d2 ff   a6
14a8 : e8 d0 f5 ee f9 04 ad f9   be
14b0 : 04 c9 16 d0 03 20 b9 04   59
14b8 : 60 a9 00 8d f9 04 a2 00   e9
14c0 : bd e5 04 f0 06 20 d2 ff   3c
14c8 : e8 d0 f5 a9 00 85 c6 a5   5d
14d0 : c6 f0 fc a2 00 bd f0 04   5b
14d8 : f0 06 20 d2 ff e8 d0 f5   a4
14e0 : a9 00 85 c6 60 12 5b 57   76
14e8 : 45 49 54 45 52 5d 92 00   ea
14f0 : 14 14 14 14 14 14 14 14   f0
14f8 : 00 00 00 00 00 00 00 00   f9
1500 : 00 00 00 00 00 00 00 00   01
1508 : 00 00 00 00 00 00 00 00   09
1510 : 00 00 00 00 00 00 00 00   11
1518 : 00 00 00 00 00 00 00 00   19
1520 : 00 00 00 00 00 00 00 00   21
1528 : a9 40 8d 04 d4 a9 0f 8d   c7
1530 : 18 d4 a2 1e a0 86 8e 00   97
1538 : d4 8c 01 d4 a2 00 a0 a0   1b
1540 : 8e 05 d4 8c 06 d4 a9 00   c5
1548 : 8d f9 04 a9 3e 20 d2 ff   38
1550 : a9 a4 20 d2 ff a9 00 8d   16
1558 : af 06 a9 00 85 c6 a5 c6   27
1560 : f0 fc ad 77 02 a2 00 dd   1a
1568 : 5d 06 f0 08 e8 e0 52 d0   86
1570 : f6 4c 77 05 4c b5 05 c9   25
1578 : 0d d0 15 ad af 06 f0 2f   36
1580 : a9 9d 20 d2 ff a9 20 20   68
1588 : d2 ff 20 f8 05 4c 08 06   60
1590 : c9 14 d0 1b ad af 06 f0   4d
1598 : 16 a9 14 20 d2 ff 20 d2   df
15a0 : ff a9 a4 20 d2 ff 20 f8   41
15a8 : 05 ce af 06 4c 5a 05 20   ad
15b0 : 36 06 4c 5a 05 ac af 06   c8
15b8 : c0 4d f0 36 ae af 06 9d   dd
15c0 : b0 06 a9 9d 20 d2 ff bd   a5
15c8 : b0 06 20 d2 ff a9 a4 20   fe
15d0 : d2 ff ee af 06 bd b0 06   70
15d8 : c9 22 d0 10 20 d2 ff a9   d4
15e0 : 14 20 d2 ff 20 d2 ff a9   a5
15e8 : a4 20 d2 ff 20 f8 05 4c   c7
15f0 : 5a 05 20 36 06 4c 5a 05   d2
15f8 : a9 41 8d 04 d4 a2 00 e8   5a
1600 : d0 fd a9 80 8d 04 d4 60   56
1608 : a9 00 85 c6 20 fd ae ad   f3
1610 : af 06 20 7d b4 a0 00 b9   3e
1618 : b0 06 91 62 c8 cc af 06   3a
1620 : d0 f5 20 8b b0 85 fb 84   94
1628 : fc a2 02 a0 02 b5 61 91   80
1630 : fb ca 88 10 f8 60 a2 0a   e6
1638 : 8e 00 d4 8e 01 d4 a9 21   6d
1640 : 8d 04 d4 a2 b0 a0 00 c8   fa
1648 : d0 fd e8 d0 f8 a9 80 8d   65
1650 : 04 d4 a6 1e a4 86 8e 00   e4
1658 : d4 8c 01 d4 60 41 42 43   ed
1660 : 44 45 46 47 48 49 4a 4b   50
1668 : 4c 4d 4e 4f 50 51 52 53   58
1670 : 54 55 56 57 58 59 5a 30   09
1678 : 31 32 33 34 35 36 37 38   68
1680 : 39 2c 2e 20 21 23 24 25   65
1688 : 26 27 28 29 2b 2d 3a 3b   ed
1690 : 2f 3f 40 2a 22 c1 c2 c3   77
1698 : c4 c5 c6 c7 c8 c9 ca cb   87
16a0 : cc cd ce cf d0 d1 d2 d3   8f
16a8 : d4 d5 d6 d7 d8 d9 da 00   df
```

**Listing 5. Einige nützliche Hilfsroutinen. Beachten Sie die Hinweise im Text.**

Sie nicht den LOAD-Befehl, sondern nur Listing 6 verwenden.

Die Routine beherbergt zwei sehr wichtige Funktionen, die bei jeder Adventure-Programmierung benötigt werden:

1. Erweiterte INPUT-Routine

Mittels SYS 1160,BE$ rufen Sie die erweiterte INPUT-Routine auf. Nach Rücksprung steht der eingegebene Satz in BE$ – natürlich kann anstelle von BE$ auch jeder andere beliebige String gesetzt werden.



2. Raumbeschreibung einlesen

Mittels SYS 1024,SA rufen Sie eine Maschinenroutine auf, die den Satzbereich einer relativen Datei von Satz SA ab bis zum nächsten Sternchen ausgibt – und das sehr, sehr schnell!

Die relative Datei muß von Basic aus per OPEN 8,8,8,»Filename« geöffnet werden, ebenso der Floppykanal mittels OPEN 15,8,15.

```
10 OPEN 1,8,0,"MS":GET#1,A$:GET#1,A$            <194>
20 FOR I=1024 TO 1718:GET#1,A$:A$=A$+CHR$(
   0):POKE I,ASC(A$):NEXT I:CLOSE 1             <102>
```

**Listing 6. Lader für Listing 5**

Nochmals sei darauf hingewiesen, daß die Routinen nur lauffähig sind, wenn der deutsche Zeichensatz verwendet wird, und daß sie nur mit den Zeilen aus Listing 6 geladen werden können.

Mit dieser Routine möchte ich mich von Ihnen verabschieden. Ich hoffe, daß Sie einige wertvolle Tips zur Programmierung von Abenteuerspielen erhalten haben.

Viel Spaß beim Programmieren! (M. Nickles/bs)

# Checksummer 64 V3

**Der Checksummer 64 V3 überprüft jede Basic-Zeile direkt nach der Eingabe, erkennt Fehleingaben sowie Vertauschungen von Ziffern und erspart eine aufwendige Fehlersuche.**

Der Checksummer 64 V3 ist ein kleines Maschinenprogramm, das Sie sofort unterrichtet, ob Sie die jeweilige Programmzeile korrekt eingegeben haben.

So gehen Sie vor:

**1.** Programm abtippen und speichern.

**2.** Starten mit RUN

**3.** Nach kurzer Zeit sehen Sie am Bildschirm: CHECKSUMMER 64, CHECKSUMMER AKTIVIERT, AUSSCHALTEN MIT POKE 1,55, ANSCHALTEN MIT POKE 1,53, READY.

**4.** Anschalten des Checksummer 64 V3 mit POKE 1,53.

**5.** Test: Geben Sie in einer freien Zeile ein: »1 REM« und drücken die RETURN-Taste. Am Bildschirm oben links sollten Sie die Prüfsumme <63> sehen.

**6.** Geben Sie ein Listing aus unserem Heft ein. Nach jeder Zeile wird die Zahl, die im Listing in Klammern < > steht, in den Bildschirm eingeblendet. Stimmen die Zahlen nicht überein, so liegt vermutlich ein Eingabefehler vor.

**Die Zahl in den Klammern, und auch die Klammern selbst, dürfen beim Abtippen nicht mit eingegeben werden!**

**7.** Der Checksummer 64 V3 bemerkt auch Vertauschungen von Zahlen und Buchstaben, aber nicht das Fehlen (oder Hinzufügen) von Leerzeichen.

**8.** Unsere Basic-Listings enthalten keine Steuerzeichen mehr. Diese werden ersetzt durch Klartext und stehen zwischen geschweiften Klammern. Deshalb sind weder die Klammern noch was dazwischen steht, abzutippen, sondern die in Tabelle 1 aufgeführten Tasten zu drücken. Auf Ihrem Bildschirm erhalten Sie dann wieder die entsprechenden Grafikzeichen.

**9.** Alle Grafikzeichen werden ebenfalls ersetzt durch unterstrichene oder überstrichene Großbuchstaben.

**Unterstrichene Buchstaben bedeuten, daß Sie die SHIFT-Taste und den angegebenen Buchstaben drücken müssen, überstrichene jedoch die Commodore-Taste mit dem Buchstaben.**

Auch hier erhalten Sie am Bildschirm das entsprechende Grafikzeichen und nicht etwa das im Listing erkennbare Zeichen.

Die Leerzeichen zwischen den einzelnen Basic-Befehlen können beim Abtippen entfallen (ohne Einfluß auf die Checksumme zu nehmen). Dies ist besonders bei speicherkritischen Programmen wichtig. Ebenso müssen Zeilen, die mehr als 80 Zeichen pro Zeile enthalten, mit den bekannten Abkürzungen für die Basic-Befehle (siehe auch das Handbuch zum C 64, Anhang D, Seite 130) eingegeben werden.

Sie können die Programme auch weiterhin ohne den Checksummer eintippen. (F. Lonczewski/gk)

**Hinweis: {13 SPACE} bedeutet 13mal die Leertaste drücken**

```
9 REM ********************************
10 PRINT"{CLR,11SPACE,RVSON}CHECKSUMMER 64
   V3{RVOFF}"
11 PRINT"{2DOWN,9SPACE}EINEN MOMENT, BITTE
   ..."
12 FOR I=828 TO 864:READ A:POKE I,A:PS=PS+
   A+1:NEXT I
13 IF PS<>5802 THEN PRINT"PRUEFSUMMENFEHLE
   R IN ZEILEN 20-22":END
14 SYS 828:PS=0:FOR I=58464 TO 58583:READ
   A:POKE I,A:PS=PS+A+1:NEXT I
15 IF PS<>16267 THEN PRINT"PRUEFSUMMENFEHL
   ER IN ZEILEN 22-30":END
16 POKE 1,53:POKE 42289,96:POKE 42290,228
17 PRINT"{4DOWN,9SPACE}CHECKSUMMER AKTIVIE
   RT."
18 PRINT"{2DOWN}AUSSCHALTEN : POKE1,55"
19 PRINT"{DOWN}ANSCHALTEN{2SPACE}: POKE1,5
   3":NEW
20 DATA 169,0,133,254,162,1,189,93,3,133,2
   55,160,0,177,254
21 DATA 145,254,136,208,249,230,255,165,25
   5,221,95,3,208,238,202
22 DATA 16,230,96,160,224,192,0,160,2,169,
   0,170,133,254,177
23 DATA 95,240,40,201,32,208,3,200,208,245
   ,133,255,138,41,7
24 DATA 170,240,14,72,165,255,24,42,105,0,
   202,208,249,133,255
25 DATA 104,170,232,165,255,24,101,254,133
   ,254,76,111,228,192,4
26 DATA 48,219,198,214,165,214,72,162,3,16
   9,32,157,1,4,189
27 DATA 212,228,32,210,255,208,12,0,92,72,
   32,201,255,170,104
28 DATA 144,1,138,96,202,16,228,166,254,16
   9,0,32,205,189,169
29 DATA 62,32,210,255,104,133,214,32,108,2
   29,169,141,32,210,255
30 DATA 76,128,164,9,60,18,19
```

© 64'er

**Der Checksummer 64 V3 erkennt auch Vertauschungen von Zahlen**

CTRL steht für Control-Taste, so bedeutet [CTRL-A], daß Sie die Control-Taste und die Taste »A« drücken müssen. Im folgenden steht:

| Steuerbefehl | Taste(n) |
|---|---|
| {DOWN} | Taste neben rechtem Shift, Cursor unten |
| {UP} | Shift-Taste & Taste neben rechtem Shift; Cursor hoch |
| {CLR} | Shift-Taste & 2. Taste ganz rechts oben |
| {INST} | Shift-Taste & Taste ganz rechts oben |
| {HOME} | 2. Taste von ganz rechts oben |
| {DEL} | Taste ganz rechts oben |
| {RIGHT} | Taste ganz rechts unten |
| {LEFT} | Shift-Taste & Taste unten rechts |
| {SPACE} | Leertaste |
| {F1} bis {F8} | Funktionstasten |
| {RETURN} | Shift-Taste & Return |
| {BLACK} | Control-Taste & 1 |
| {WHITE} | Control-Taste & 2 |
| {RED} | Control-Taste & 3 |
| {CYAN} | Control-Taste & 4 |
| {PURPLE} | Control-Taste & 5 |
| {GREEN} | Control-Taste & 6 |
| {BLUE} | Control-Taste & 7 |
| {YELLOW} | Control-Taste & 8 |
| {RVSON} | Control-Taste & 9 |
| {RVOFF} | Control-Taste & 0 |
| {ORANGE} | Commodore-Taste & 1 |
| {BROWN} | Commodore-Taste & 2 |
| {LIG.RED} | Commodore-Taste & 3 |
| {GREY 1} | Commodore-Taste & 4 |
| {GREY 2} | Commodore-Taste & 5 |
| {LIG.GREEN} | Commodore-Taste & 6 |
| {LIG.BLUE} | Commodore-Taste & 7 |
| {GREY 3} | Commodore-Taste & 8 |

**Tabelle 1. Die Steuerbefehle in den Listings**

# MSE – Abtippen sicher und leicht gemacht

**Ähnlich wie der »Checksummer« ist auch der MSE ein leicht zu bedienendes Hilfsmittel bei der Eingabe von Listings, diesmal jedoch bei reinen Maschinensprache-Programmen.**

Im Gegensatz zum »Checksummer« aber ist die Eingabe nicht ohne den MSE möglich. Der MSE verringert die Tipparbeit um ein Drittel und schließt Fehleingaben vollkommen aus. Außerdem können Sie die Werte blind eingeben, ohne andauernd auf den Bildschirm schauen zu müssen. Dies wird durch akustische Meldungen realisiert.

MSE ist ein Maschinenspracheditor, mit dem ein Vertippen ausgeschlossen ist. Eine abgetippte Zeile wird nur angenommen, wenn sie richtig ist. Eine Checksumme am Ende jeder Zeile prüft, ob die richtigen Werte in der richtigen Zeile an der richtigen Stelle stehen. Wenn nicht, ertönt ein Warnsignal, und man beseitigt den Fehler.

War die Zeile korrekt, erklingt ein Gong, und die nächste Zeilennummer wird ausgegeben. Damit ist also auch »blindes« Eintippen möglich; Sie können sich voll auf den Text konzentrieren.

### So arbeitet man mit MSE

Laden und starten Sie MSE. Zuerst wird der Programmname und die Start- und Endadresse erfragt. **Diese Angaben entnehmen Sie dem Kopf des jeweiligen abgedruckten Listings.** MSE meldet sich dann mit der Zeilennummer der ersten Zeile.

Wenn Sie die Zeile richtig eingegeben haben, erscheint die nächste Zeilennummer und so weiter bis zum Ende. Zum Schluß wird das fertige Programm mit »CTRL-S« auf Diskette oder Kassette abgespeichert. Dazu sind keine weiteren Angaben mehr erforderlich. Das Programm kann dann ganz normal wieder geladen und gestartet werden. Wenn Sie nicht alles auf einmal tippen wollen, können Sie jederzeit unterbrechen und den eingetippten Teil mit »CTRL-S« abspeichern. Wollen Sie weiterarbeiten, laden und starten Sie MSE wieder.

Geben Sie auf die Frage nach der Startadresse aber jetzt »L« ein, um Ihr Teilprogramm zu laden. Jetzt können Sie mit »CTRL-N« die Adresse eingeben, an der Sie weitertippen müssen. Wenn Sie sich nicht gemerkt haben, wie weit Sie gekommen sind, geben Sie nach dem Laden »CTRL-M« ein.

Auf die Frage nach der Startadresse antworten Sie mit der Anfangsadresse, die links in der Kopfzeile auf dem Bildschirm steht. Nun wird Ihr Programm aufgelistet. Mit »SPACE« wird das Listen fortgesetzt, mit »STOP« abgebrochen. Das Ende Ihres Programmteils erkennen Sie sehr einfach daran, daß nur noch der Wert »AA« in der Zeile steht. Die Adresse dieser Zeile müssen Sie anschließend mit »CTRL-N« eingeben. Das Programm ist nur mit »STOP/RESTORE« zu verlassen. Speichern Sie aber vorher unbedingt immer Ihren Text ab.

### Hinweise zum Abtippen

Vor dem Abtippen oder späteren Wiederladen des MSE-**Laders** müssen Sie unbedingt folgende Zeile eingeben:
**POKE 43,1: POKE 44,32: POKE 8192,0: NEW**

Den MSE-**Lader** brauchen Sie nur einmal. Nach erfolgreichem Abtippen und Starten mit RUN geht der Lader verloren und es wird das endgültige Programm MSE V1.0 erzeugt. So gehen Sie vor:

Starten Sie das Programm mit RUN. Fehlerhafte Zeilen werden angezeigt und müssen korrigiert werden, bis der Lader zum »READY« durchläuft. Jetzt müssen Sie das fertige MSE-Programm speichern. Dazu brauchen Sie nur »RETURN« zu drücken, weil die erforderlichen Angaben schon auf dem Bildschirm stehen. (Kassettenbesitzer müssen in Zeile 343 die letzte Zahl in »1« abändern.) Ab jetzt können Sie »MSE V1.0« direkt, also ohne den DATA-Lader, benutzen. MSE V1.0 wird ganz normal mit »,8« geladen (keine POKEs notwendig).

(N. Mann/D. Weineck/gk)



**MSE-Befehle:**

| | |
|---|---|
| DEL | löscht die letzte Eingabe. |
| CTRL-S | speichert das eingetippte Programm ab. |
| L oder CTRL-L | lädt ein Programm. Start- und Endadresse werden automatisch ermittelt. |
| CTRL-M | listet den Speicherinhalt. Abbruch mit STOP-Taste, weiter mit Leertaste. |
| CTRL-N | erlaubt die Eingabe einer neuen Adresse zum Weitertippen. |
| CTRL-P | gibt ein MSE-Listing auf dem Drucker aus. |

```
100 REM *****************************        <091>
110 REM *                           *        <159>
120 REM *        M S E   LADER      *        <206>
130 REM *                           *        <179>
220 REM *****************************        <211>
230 REM                                      <036>
240 DIM H(75): FOR I=0 TO 9                  <113>
250 H(48+I)=I: H(65+I)=I+10:NEXT             <041>
260 FOR I=2048 TO 3755 : READ A$             <198>
270 H=ASC(LEFT$(A$,1)):L=ASC(RIGHT$(A$,1))   <199>
280 D=H(H)*16+H(L):S=S+D:POKE I,D            <219>
290 A=A+1:IF A<20 THEN NEXT:A=-1             <141>
300 PRINT " ZEILE:";1000+Z;                  <011>
310 READ V :Z=Z+1:IF V=S THEN 330            <218>
320 PRINT"PRUEFSUMMENFEHLER !":STOP          <138>
330 IF A<0 THEN 341                          <221>
340 S=0:A=0:PRINT:NEXT                       <046>
341 PRINT"{CLR}P043,1:P044,8:P045,172:P046
    ,14                                      <010>
342 POKE 631,19:POKE 632,13:POKE 633,13:PO
    KE 198,3                                 <?49>
343 PRINT"{3DOWN}SAVE"CHR$(34)"MSE V1.0"CH
    R$(34)",8                                <171>
344 END                                      <092>
1000 DATA 00,0B,08,0A,00,9E,32,30,36,31,00
     ,00,00,A2,08,A9,36,85,A4,A9, 1247       <119>
1001 DATA 08,85,A5,A9,00,85,A6,A9,B0,85,A7
     ,A0,00,B1,A4,91,A6,C8,D0,F9, 2888       <054>
1002 DATA E6,A5,E6,A7,CA,D0,F2,A9,36,85,01
     ,4C,00,B0,20,D1,B1,A9,06,8D, 2787       <144>
1003 DATA 21,D0,A9,03,8D,20,D0,8D,86,02,A0
     ,B3,A9,74,20,FF,B1,A0,B3,A9, 2667       <237>
1004 DATA B9,20,FF,B1,A0,00,20,CF,FF,99,01
     ,02,C8,C9,0D,D0,F5,88,F0,D2, 2912       <217>
1005 DATA C0,0F,90,02,A0,0E,8C,00,02,20,EA
     ,B1,A0,B3,A9,CF,20,FF,B1,20, 2323       <013>
1006 DATA 8E,B4,85,FC,85,62,20,8E,B4,85,FB
     ,85,61,20,A7,B4,D0,20,A0,B3, 2864       <199>
1007 DATA A9,E5,20,FF,B1,20,8E,B4,85,60,20
     ,8E,B4,85,5F,20,A7,B4,D0,0A, 2624       <091>
```

**Der MSE zum bequemen Abtippen von Maschinenprogrammen**

```
1008 DATA A5,61,C5,5F,A5,62,E5,60,90,06,20
     ,43,B3,4C,3A,B0,A9,AA,A0,00, 2379      <167>
1009 DATA 91,FB,E6,FB,D0,02,E6,FC,20,3F,B2
     ,90,EF,4C,FB,B4,A2,02,86,58, 3118      <152>
1010 DATA A9,A6,A0,9D,20,F2,B1,20,E4,FF,F0
     ,FB,C9,30,90,0C,C9,47,B0,08, 2970      <231>
1011 DATA C9,3A,90,0B,C9,41,B0,07,C9,14,D0
     ,0F,4C,0B,B1,20,D2,FF,A6,58, 2322      <121>
1012 DATA 95,F7,C6,58,D0,D2,60,AE,8D,02,F0
     ,26,C9,0C,D0,03,4C,0B,B6,C9, 2685      <057>
1013 DATA 13,D0,03,4C,8B,B5,C9,0D,D0,03,4C
     ,BA,B4,C9,10,D0,03,4C,68,B5, 2282      <225>
1014 DATA C9,0E,D0,06,20,5F,B4,4C,64,B1,4C
     ,92,B0,A5,F9,20,02,B1,0A,0A, 2132      <208>
1015 DATA 0A,0A,85,F9,A5,F8,20,02,B1,05,F9
     ,60,C9,3A,90,02,69,08,29,0F, 1950      <092>
1016 DATA 60,A6,59,E0,08,90,1F,A6,58,E0,02
     ,B0,06,20,D2,FF,4C,8E,B0,C6, 2509      <188>
1017 DATA 59,A0,14,A9,92,20,F2,B1,CA,D0,FA
     ,84,57,68,68,4C,8B,B1,A6,D3, 2891      <197>
1018 DATA E0,08,B0,03,4C,92,B0,20,D2,FF,A6
     ,58,E0,02,90,09,C6,59,20,D2, 2468      <049>
1019 DATA FF,C6,58,D0,F9,4C,8E,B0,48,4A,4A
     ,4A,4A,20,59,B1,68,29,0F,C9, 2419      <035>
1020 DATA 0A,90,02,69,06,69,30,4C,D2,FF,A2
     ,FC,9A,20,D1,B1,20,48,B2,20, 2261      <073>
1021 DATA EA,B1,20,9F,B2,A5,FC,20,4E,B1,A5
     ,FB,20,4E,B1,20,ED,B1,A9,3A, 2860      <148>
1022 DATA A0,20,20,F2,B1,A9,00,85,59,20,8E
     ,B0,20,ED,B1,A4,59,20,EF,B0, 2530      <233>
1023 DATA 91,FB,C8,84,59,C0,08,90,EC,20,10
     ,B2,A9,12,20,D2,FF,20,8E,B0, 2657      <105>
1024 DATA 20,EF,B0,C5,FF,F0,0D,20,43,B3,A9
     ,14,A0,14,20,F2,B1,4C,A2,B1, 2665      <034>
1025 DATA A9,92,20,D2,FF,20,33,B2,20,E0,B2
     ,20,3F,B2,90,9F,4C,8B,B5,A9, 2648      <123>
1026 DATA 93,20,D2,FF,A2,00,A9,03,9D,00,D8
     ,9D,00,D9,9D,00,DA,9D,00,DB, 2476      <237>
1027 DATA E8,D0,EF,60,A9,0D,2C,A9,20,4C,D2
     ,FF,20,D2,FF,98,4C,D2,FF,20, 2965      <160>
1028 DATA E4,FF,F0,FB,60,84,5D,85,5C,A0,00
     ,B1,5C,F0,06,20,D2,FF,C8,D0, 3100      <077>
1029 DATA F6,60,A5,FB,85,5A,A0,00,84,5B,B1
     ,FB,18,65,5A,85,5A,90,02,E6, 2606      <156>
1030 DATA 5B,06,5A,26,5B,C8,C0,08,90,EC,A5
     ,5A,65,5B,85,FF,60,18,A5,FB, 2467      <219>
1031 DATA 69,08,85,FB,90,02,E6,FC,60,A5,FB
     ,C5,5F,A5,FC,E5,60,60,A0,B3, 3106      <183>
1032 DATA A9,FB,20,FF,B1,A0,01,B9,00,02,20
     ,D2,FF,CC,00,02,C8,90,F4,A9, 2692      <098>
1033 DATA 10,ED,00,02,AA,20,ED,B1,CA,D0,FA
     ,A5,62,20,4E,B1,A5,61,20,4E, 2453      <236>
1034 DATA B1,20,ED,B1,A5,60,20,4E,B1,A5,5F
     ,20,4E,B1,A9,9F,20,D2,FF,20, 2575      <038>
1035 DATA EA,B1,24,5E,10,01,60,A9,12,20,D2
     ,FF,A2,28,20,ED,B1,CA,D0,FA, 2646      <161>
1036 DATA A9,92,4C,D2,FF,A5,D6,C9,16,B0,01
     ,60,A9,A0,85,A4,A9,78,85,A6, 2945      <204>
1037 DATA A9,04,85,A5,85,A7,A2,13,A0,27,B1
     ,A4,91,A6,88,10,F9,CA,F0,19, 2671      <208>
1038 DATA 18,A5,A4,69,28,85,A4,90,02,E6,A5
     ,18,A5,A6,69,28,85,A6,90,E0, 2503      <251>
1039 DATA E6,A7,4C,B6,B2,A9,91,4C,D2,FF,A9
     ,0F,8D,18,D4,A9,00,8D,05,D4, 2776      <000>
1040 DATA A9,F7,8D,06,D4,A9,11,8D,04,D4,A9
     ,32,8D,01,D4,A9,00,8D,00,D4, 2413      <126>
1041 DATA A0,80,20,09,B3,A9,10,8D,04,D4,60
     ,A2,FF,CA,D0,FD,88,D0,F8,60, 2914      <240>
1042 DATA A9,0F,8D,18,D4,A9,2D,8D,05,D4,A9
     ,A5,8D,06,D4,A9,21,8D,04,D4, 2385      <119>
1043 DATA A9,07,8D,01,D4,A9,05,8D,00,D4,A0
     ,FF,20,09,B3,A9,20,8D,04,D4, 2250      <078>
1044 DATA A9,00,8D,01,D4,8D,00,D4,60,38,20
     ,F0,FF,8A,48,98,48,18,A0,06, 2179      <175>
1045 DATA A2,18,20,F0,FF,A0,B4,A9,0A,20,FF
     ,B1,20,12,B3,20,E4,FF,F0,FB, 2931      <093>
1046 DATA A2,1D,A9,14,20,D2,FF,CA,D0,FA,68
     ,A8,68,AA,18,4C,F0,FF,0D,0D, 2704      <088>
1047 DATA 0D,20,20,20,20,20,20,20,4D,41,53
     ,43,48,49,4E,45,4E,53,50,52, 1144      <216>
1048 DATA 41,43,48,45,20,2D,20,45,44,49,54
     ,4F,52,20,0D,0D,20,20,20,20, 1023      <038>
1049 DATA 20,20,20,20,56,4F,4E,20,4E,2E,4D
     ,41,4E,4E,20,26,20,44,2E,57, 1128      <206>
1050 DATA 45,49,4E,45,43,4B,00,0D,0D,0D,20
     ,20,20,50,52,4F,47,52,41,4D, 1102      <117>
1051 DATA 4D,4E,41,4D,45,20,3A,20,00,0D,0D
     ,20,20,20,53,54,41,52,54,41, 1073      <095>
1052 DATA 44,52,45,53,53,45,20,3A,20,24,00
     ,0D,0D,20,20,20,45,4E,44,41, 1014      <129>
1053 DATA 44,52,45,53,53,45,20,20,20,3A,20
     ,24,00,92,05,20,50,52,4F,47, 1171      <217>
1054 DATA 52,41,4D,4D,20,3A,20,00,12,20,20
     ,2A,2A,2A,20,46,41,4C,53,43, 1024      <027>
1055 DATA 48,45,20,45,49,4E,47,41,42,45,20
     ,2A,2A,2A,20,20,92,00,0D,0D, 1058      <098>
1056 DATA 2A,2A,2A,20,45,4E,44,45,20,2A,2A
     ,2A,00,13,05,20,20,12,44,92, 920       <148>
1057 DATA 49,53,4B,20,4F,44,45,52,20,12,54
     ,92,41,50,45,0D,00,13,20,20, 1151      <035>
1058 DATA 49,2F,4F,20,2D,20,46,45,48,4C,45
     ,52,00,20,D1,B1,20,48,B2,A0, 1606      <012>
1059 DATA B3,A9,CF,20,FF,B1,20,8E,B4,85,FC
     ,20,8E,B4,85,FB,C5,61,A5,FC, 3207      <251>
1060 DATA E5,62,90,23,A5,FB,C5,5F,A5,FC,E5
     ,60,B0,19,20,A7,B4,D0,14,60, 2860      <112>
1061 DATA 20,A7,B4,F0,0C,85,F9,20,A7,B4,F0
     ,05,85,F8,4C,EF,B0,68,68,20, 2749      <088>
1062 DATA 43,B3,4C,5F,B4,20,CF,FF,C9,4C,D0
     ,09,20,D1,B1,20,48,B2,4C,0B, 2372      <046>
1063 DATA B6,C9,0D,60,A9,00,85,5E,20,5F,B4
     ,20,EA,B1,20,0D,B5,24,5E,30, 2042      <120>
1064 DATA 05,20,E4,FF,F0,FB,20,E1,FF,F0,26
     ,20,9F,B2,24,5E,10,09,20,4E, 2435      <198>
1065 DATA B5,20,0D,B5,20,60,B5,20,33,B2,20
     ,3F,B2,90,D7,A0,B4,A9,28,20, 2190      <207>
1066 DATA FF,B1,20,E4,FF,C9,0D,D0,F9,A9,00
     ,85,5E,A5,61,85,FB,A5,62,85, 3056      <240>
1067 DATA FC,20,E0,B2,4C,64,B1,A5,FC,20,4E
     ,B1,A5,FB,85,FF,20,4E,B1,A9, 3003      <221>
1068 DATA 20,A0,3A,20,F2,B1,A0,00,20,ED,B1
     ,B1,FB,20,4E,B1,C8,C0,08,90, 2566      <070>
1069 DATA F3,20,ED,B1,24,5E,30,03,A9,12,2C
     ,A9,20,20,D2,FF,20,10,B2,A5, 2190      <059>
1070 DATA FF,20,4E,B1,A9,92,20,D2,FF,4C,EA
     ,B1,A9,FF,85,B8,85,B9,A9,04, 3073      <029>
1071 DATA 85,BA,20,C0,FF,A2,FF,4C,C9,FF,20
     ,CC,FF,A9,FF,4C,C3,FF,20,5F, 3315      <189>
1072 DATA B4,A9,80,85,5E,20,4E,B5,20,48,B2
     ,A2,24,A9,2D,20,D2,FF,CA,D0, 2596      <111>
1073 DATA FA,20,EA,B1,20,EA,B1,20,60,B5,4C
     ,C1,B4,20,B8,B5,A6,5F,A4,60, 2812      <015>
1074 DATA A9,61,20,D8,FF,B0,0A,20,B7,FF,29
     ,BF,D0,03,4C,FB,B4,A9,01,20, 2577      <201>
1075 DATA C3,FF,20,68,B6,A0,B4,A9,4F,20,FF
     ,B1,20,F9,B1,4C,FB,B4,20,68, 2921      <237>
1076 DATA B6,A9,37,A0,B4,20,FF,B1,20,F9,B1
     ,A2,08,C9,44,F0,06,A2,01,C9, 2717      <213>
1077 DATA 54,D0,F1,A9,01,A8,20,BA,FF,A0,00
     ,E0,01,F0,1A,A9,40,8D,20,02, 2403      <101>
1078 DATA A9,3A,8D,21,02,B9,01,02,99,22,02
     ,C8,CC,00,02,90,F4,C8,C8,D0, 2182      <127>
1079 DATA 0C,B9,01,02,99,20,02,C8,CC,00,02
     ,D0,F4,98,A2,20,A0,02,4C,BD, 2018      <025>
1080 DATA FF,20,B8,B5,A5,BA,C9,08,90,33,A6
     ,B9,86,57,A9,01,20,C3,FF,A9, 2800      <022>
1081 DATA 60,85,B9,20,C0,FF,B0,28,A5,BA,20
     ,B4,FF,A5,B9,20,96,FF,20,A5, 2911      <053>
1082 DATA FF,85,61,A5,90,4A,4A,B0,13,20,A5
     ,FF,85,62,20,AB,FF,A5,57,85, 2663      <214>
1083 DATA B9,A9,00,20,D5,FF,90,03,4C,A3,B5
     ,86,5F,84,60,A5,BA,C9,01,D0, 2639      <131>
1084 DATA 0A,AD,3D,03,85,61,AD,3E,03,85,62
     ,4C,FB,B4,A9,13,20,D2,FF,A2, 2300      <120>
1085 DATA 1C,20,ED,B1,CA,D0,FA,60, 1230      <214>
```



**MSE (Schluß). Dieses Listing können Sie (müssen aber nicht) mit dem Checksummer 64 V3 in diesem Heft eingeben.**

# Odyssey – Kampf mit der Wildnis

**Sie sind nach einem Schiffsunglück auf einer einsamen Insel gestrandet. Finden Sie den Weg zurück in die Zivilisation und entkommen Sie dem Abenteuer Wildnis.**

Odyssey ist ein deutsches Textadventure, das von Hans-Günther Schreck, einem Freund von mir, geschrieben worden ist. Hans begann mit dem Programmieren dieses Spieles, als ich gerade mit der Fertigstellung meiner »Gordon Saga 1« beschäftigt war. Ich erwähne dies an dieser Stelle, da mir Hans bei der Fertigstellung von Gordon Saga viel geholfen hat – und zwar als Testspieler. Ich beobachtete ihn stundenlang beim Spielen von Gordon Saga und machte mir Notizen, wo das Spiel zu einfach oder zu schwer war und wo es am Parser mangelte. Ich kann ein solches Testen jedem Adventure-Autor wärmstens empfehlen. Ein Adventure zu programmieren nimmt viel Zeit in Anspruch. Wer eines fertiggestellt hat, kennt meist jede Szene in- und auswendig und hat keine Lust mehr zu langem Probespielen, da man dann meist auch schon an neue Projekte denkt. Läßt man jedoch einen Bekannten, der nichts mit dem Spiel zu tun hat, einmal probespielen, so merkt man schnell, ob am Programm noch etwas zu verbessern ist. Hier ein Beispiel:

In der Handlung der Gordon Saga muß an bestimmten Stellen eine Tür geöffnet werden. Ich hatte hierzu als Syntax »ÖFFNE DIE TÜR« vorgesehen. Einige Testspieler gaben »ÖFFNE DIE TÜRE« ein und konnten nicht verstehen, warum dieser Befehl nicht verstanden wurde, da es sich bei »TÜRE« um einen offensichtlich verbreiteten Rechtschreibfehler handelt. Ich nahm im Wortschatz für »TÜR« also auch »TÜRE« auf. Ein anderer gab »MACHE DIE TÜR AUF« ein. Auch hier verbesserte ich den Parser. Nach langen Teststunden war Gordon Saga endlich fertig. Besonders ratsam ist es auch, als Testpersonen Leute herzunehmen, die zuvor noch kein Adventure gespielt haben. Dann merkt man am besten, wie gut der Parser des Spieles ist.

## Die Spielhandlung

Nun aber zu »Odyssey«. Das Spiel ist komplett in Basic geschrieben, die Grafik besteht aus dem Commodore-Standardzeichensatz, ist jedoch so gut, daß man dies bei manchen Bildern einfach nicht glauben kann. Der Parser entspricht keinesfalls dem in diesem Sonderheft gesetzten Standard. Trotzdem werden ganze deutsche Sätze verstanden, was deutlich über dem Standard gewöhnlicher Abtipp-Adventures liegt.

Sie sind nach einem Schiffsunglück auf einer einsamen Insel gestrandet und müssen sich in der Wildnis herumschlagen, Raubtiere bekämpfen oder sich mit Eingeborenen herumschlagen. Dabei begegnen Sie noch vielen anderen Geheimnissen wie Schätzen oder Botschaften. Das Spiel zeichnet sich durch viel Spielwitz und Originalität aus. Mehr wird nicht verraten – einfach abtippen und spielen.

(Michael Nickles/dm)

```
0 PRINT"{CLR}":POKE 53280,0:POKE 53281,0:F
  =2:GOSUB 20000:POKE 198,0:GOSUB 750       <148>
10 DIM T$(80):DIM RA$(20,10):WL=3:TI$="000
   000"                                     <177>
20 REM *** GEGENSTAENDE ******************
   ************************************     <063>
22 G$(0)=" ":G$(1)="MESSER":G$(2)="METALLB
   EHAELTER":G$(3)="BUCH":G$(4)="BEIL"      <239>
24 G$(5)="FRUECHTE":G$(6)="VERBANDSZEUG":G
   $(7)="SCHLANGENSERUM"                    <196>
26 G$(8)="EDELSTEINE":G$(9)="FALTBOOT"      <019>
36 PRINT"{CLR}"CHR$(14):GOSUB 25000         <062>
80 REM *** SPRUNG IN RAUM ****************
   ************************************     <030>
90 GOTO 1000                                <030>
100 IF UN=1 THEN GOTO 200                   <195>
101 PRINT"{WHITE}"                          <112>
102 ZV=RND(1):IF ZV<0.5 THEN PRINT"WAS NUN
     ?":GOTO 110                            <217>
103 IF ZV>0.5 AND ZV<0.80 THEN PRINT"WAS J
    ETZT ?":GOTO 110                        <028>
105 PRINT"UND JETZT ?"                      <099>
110 PRINT"{DOWN,UP}":X$="":PRINT CHR$(114) <123>
120 GET B$                                  <082>
125 IF ZA=1 THEN IF TI-D>LI THEN T=1:R=0:P
    $="":GOTO 300                           <073>
127 IF B$=""THEN 120                        <066>
130 IF ASC(B$)=13 THEN 198                  <178>
140 IF ASC(B$)=20 AND X$<>""THEN 160        <109>
142 IF ASC(B$)=32 THEN 150                  <059>
145 IF ASC(B$)<65 OR ASC(B$)>90 THEN 120    <028>
150 X$=X$+B$:GOTO 180                       <144>
160 X$=LEFT$(X$,LEN(X$)-1)                  <019>
170 IF LEN(X$)<41 AND LEN(X$)>37 THEN PRIN
    T"{2UP}"X$+CHR$(114)+CHR$(32):UA=1:GOT
    O 120                                   <122>
175 GOTO 190                                <247>
180 IF LEN(X$)<40 AND UA=0 THEN PRINT"{UP}
    "X$+CHR$(114):UA=0:GOTO 120             <217>
185 PRINT"{2UP}"X$+CHR$(114):GOTO 194       <255>
190 IF LEN(X$)<41 AND UA=0 THEN PRINT"{UP}
    "X$+CHR$(114)+CHR$(20):GOTO 194         <040>
192 PRINT"{2UP}"X$+CHR$(114)+CHR$(20):GOTO
     194                                    <064>
194 IF LEN(X$)<79 THEN 120                  <223>
195 GET B$:IF B$<>CHR$(20)AND B$<>CHR$(13)
    THEN 195                                <216>
196 PRINT"{2UP}":GOTO 130                   <099>
198 IF LEN(X$)<39 THEN PRINT"{UP}"X$+CHR$(
    46):GOTO 200                            <018>
199 PRINT"{2UP}"X$+CHR$(46)                 <105>
200 REM *** STRINGZERLEGUNG ***************
    *********************************** <164>
210 UN=0:WO=0:SZ=0:T$(0)=""                 <070>
217 IF X$=""AND UN=0 THEN GOTO 713          <172>
220 FOR X=1 TO LEN(X$)                      <176>
230 :    IF MID$(X$,X,1)<>" "AND WO=0 THEN
    WO=X                                    <207>
235 : IF MID$(X$,X,1)=" "AND WO<>0 THEN T$
    (SZ)=MID$(X$,WO,X-WO):WO=0:SZ=SZ+1:GOT
```

Listing. »Odyssey« (Fortsetzung auf Seite 97)

**Listing »Odyssey« (Fortsetzung)**

```
    O 280                                          <189>
240 :IF X=LEN(X$)AND WO<>0 THEN T$(SZ)=MID
    $(X$,WO,X-WO+1):SZ=SZ+1                        <246>
270 NEXT:GOTO 300                                  <013>
280 :   IF T$(SZ-1)="UND"THEN X$=RIGHT$(X$
    ,LEN(X$)-X):UN=1:GOTO 300                      <240>
290 GOTO 270                                       <084>
300 REM *** WORTANALYSE ******************
    ************************************** <093>
305 S=0:V=0:R=0:G=0:Q=0                            <196>
310 FOR I=0 TO SZ-1                                <000>
315 IF T$(0)=""THEN GOSUB 704:RETURN               <234>
320 :   P$=T$(I):F=ASC(P$)-64                      <155>
330 :   ON F GOTO 350,360,370,380,390,400,
    410,420,430,440,450,460,470                    <098>
340 :   ON F-13 GOTO 480,490,500,510,520,5
    26,545,550,560,570,580,590,600                 <207>
345 NEXT                                           <101>
346 IF V=0 AND S=0 AND G=0 AND R=0 THEN GO
    SUB 702                                        <113>
347 GOSUB 800                                      <093>
349 RETURN                                         <153>
350 REM *** A-BER                                  <209>
351 IF P$="AST"THEN S=3:GOTO 345                   <109>
352 IF P$="ALLES"THEN S=1                          <217>
353 IF P$="ASCHE"THEN S=7                          <113>
359 GOTO 345                                       <017>
360 IF P$="BETRETE"OR P$="BETRITT"THEN V=2
    7                                              <152>
361 IF LEFT$(P$,4)="BAUM"THEN S=2:GOTO 345 <136>
362 IF P$="BUCH"THEN G=3:S=10                      <071>
363 IF P$="BEIL"THEN G=4                           <159>
364 IF P$="BEHAELTER"THEN G=2                      <113>
365 IF P$="BRING"THEN V=3                          <238>
366 IF P$="BERUEHRE"THEN V=23                      <218>
367 IF P$="BILD"THEN S=25:GOTO 345                 <095>
368 IF P$="BEHANDLE"OR P$="BEHANDELE"THEN
    V=26                                           <069>
369 IF P$="BOOT"THEN G=9                           <093>
370 REM *** C-BER                                  <101>
379 GOTO 345                                       <037>
380 REM *** D-BER                                  <176>
381 IF P$="DREHE"THEN V=15                         <055>
382 IF P$="DRUECKE"THEN V=23                       <112>
389 GOTO 345                                       <047>
390 REM *** E-BER                                  <250>
391 IF P$="EDELSTEINE"THEN G=8                     <141>
395 IF P$="ESSEN"THEN V=5                          <184>
396 IF P$="ENTZUENDE"THEN V=25                     <001>
399 GOTO 345                                       <057>
400 IF P$="FAHRE"THEN V=31                         <098>
401 IF P$="FRUECHTE"THEN G=5:S=11                  <075>
402 IF P$="FASS"THEN S=8                           <120>
403 IF P$="FELSWAND"THEN S=14                      <078>
404 IF P$="FUELLE"THEN V=11                        <127>
405 IF P$="FAELLE"THEN V=14                        <120>
406 IF P$="FUENF"THEN V=29                         <125>
407 IF P$="FALTBOOT"THEN G=9                       <147>
408 IF P$="FELSENINSEL"THEN S=32                   <060>
409 GOTO 345                                       <067>
410 REM *** G-BER                                  <142>
411 IF P$="GOLDSCHALE"THEN S=13                    <051>
412 IF P$="GORILLA"THEN S=19                       <023>
413 IF P$="GOETZENBILD"THEN S=25                   <169>
414 IF P$="GRABE"THEN V=30                         <177>
415 IF P$="GEHE"THEN V=27                          <079>
419 GOTO 345                                       <077>
420 REM *** H-BER                                  <217>
421 IF P$="H"OR P$="HINAUF"THEN R=5                <052>
422 IF P$="HINUNTER"THEN R=6                       <160>
423 IF P$="HAND"THEN S=20                          <117>
424 IF P$="HAENGE"THEN V=17                        <120>
425 IF P$="HINUEBER"THEN G=10                      <255>
426 IF P$="HANGLE"OR P$="HANGELE"THEN V=22 <134>
427 IF P$="HUETTE"THEN S=30                        <117>
429 GOTO 345                                       <087>
430 REM *** I-BER                                  <035>
431 IF P$="INSEL"THEN S=32                         <106>
435 IF P$="I"OR P$="INVENTUR"THEN S=4              <051>
436 IF P$="ISS"THEN V=5                            <240>
439 GOTO 345                                       <097>
440 REM *** J-BER                                  <109>
441 IF P$="JAGUAR"THEN S=29                        <211>
449 GOTO 345                                       <107>
450 REM *** K-BER                                  <183>
451 IF P$="KISTE"THEN S=15                         <130>
459 GOTO 345                                       <117>
460 REM *** L-BER                                  <002>
461 IF P$="LIANE"THEN S=18                         <215>
465 IF P$="LEGE"THEN V=2                           <203>
466 IF P$="LEERE"THEN V=15                         <171>
467 IF P$="LIES"THEN V=8                           <008>
469 GOTO 345                                       <127>
470 REM *** M-BER                                  <076>
471 IF P$="MESSER"THEN G=1                         <186>
472 IF P$="METALLBEHAELTER"THEN G=2                <048>
473 IF P$="METALLKISTE"THEN S=11                   <223>
474 IF P$="MAENNCHEN"THEN S=22                     <057>
475 IF P$="MACHE"THEN V=12                         <254>
479 GOTO 345                                       <137>
480 REM *** N-BER                                  <150>
481 IF P$="N"OR LEFT$(P$,3)="NOR"THEN R=1          <138>
485 IF P$="NIMM"THEN V=1                           <143>
489 GOTO 345                                       <147>
490 REM *** O-BER                                  <224>
491 IF P$="O"OR LEFT$(P$,3)="OST"THEN R=4          <169>
492 IF P$="OEFFNE"THEN V=7                         <141>
499 GOTO 345                                       <157>
500 REM *** P-BER                                  <043>
501 IF P$="PULVER"THEN S=21                        <052>
502 IF P$="PADDELE"THEN V=31                       <203>
509 GOTO 345                                       <167>
510 REM *** Q-BER                                  <117>
519 GOTO 345                                       <179>
520 REM *** R-BER                                  <193>
521 IF P$="R"OR P$="RUNTER"THEN R=6                <110>
522 IF P$="RUTSCHE"THEN V=24                       <022>
523 IF P$="RAMPE"THEN S=26                         <047>
524 IF P$="RUDERE"THEN V=31                        <021>
525 GOTO 345                                       <185>
526 IF P$="SPRITZE"THEN V=32                       <006>
527 IF LEFT$(P$,6)="SPALTE"THEN S=33               <221>
530 IF P$="S"OR LEFT$(P$,4)="SUED"THEN R=2 <153>
531 IF LEFT$(P$,4)="SING"THEN V=19                 <196>
532 IF P$="SCHLANGENSERUM"OR RIGHT$(P$,5)=
    "SERUM"THEN G=7                                <093>
533 IF P$="SCHAU"THEN S=5                          <140>
534 IF P$="SCHALE"THEN S=13                        <215>
535 IF P$="STRAND"THEN S=6                         <059>
536 IF P$="STRAUCH"THEN S=12                       <225>
537 IF LEFT$(P$,6)="SCHUET"THEN V=9                <179>
538 IF P$="SCHLIESSE"THEN V=10                     <145>
539 IF P$="SCHUETTE"THEN V=11                      <212>
540 IF P$="SCHIESSPULVER"THEN S=21                 <142>
541 IF LEFT$(P$,6)="SPRING"THEN V=13               <091>
542 IF P$="SAND"THEN KG=1                          <216>
543 IF P$="SPINNE"THEN S=23                        <053>
544 GOTO 345                                       <213>
545 REM *** T-BER                                  <091>
546 IF P$="TISCH"THEN S=9                          <218>
547 IF P$="TOETE"THEN V=20                         <011>
549 GOTO 345                                       <209>
550 REM *** U-BER                                  <160>
555 IF P$="UNTERSUCHE"THEN V=6                     <227>
559 GOTO 345                                       <219>
560 REM *** V-BER                                  <234>
561 IF P$="VERBANDSZEUG"THEN G=6                   <223>
562 IF P$="VOGELSPINNE"THEN S=23                   <138>
563 IF P$="VORSPRUNG"OR P$="VORSPRUENGE"TH
    EN S=24                                        <198>
564 IF P$="VERBINDE"THEN V=26                      <078>
565 IF LEFT$(P$,7)="VERLIER"THEN V=2               <185>
566 IF P$="VERLASSE"THEN V=28                      <139>
567 IF P$="VERNICHTE"THEN V=20                     <030>
569 GOTO 345                                       <229>
570 REM *** W-BER                                  <052>
571 IF P$="W"OR LEFT$(P$,4)="WEST"THEN R=3 <177>
572 IF LEFT$(P$,5)="WARTE"THEN V=4                 <166>
573 IF P$="WAND"THEN S=14                          <023>
574 IF P$="WILDSTRAUCH"THEN S=12                   <090>
575 IF P$="WIRF"THEN V=11                          <142>
576 IF P$="WENDE"THEN V=15                         <122>
577 IF P$="WEG"THEN S=27                           <010>
578 IF P$="WASSER"THEN S=31                        <246>
579 GOTO 345                                       <239>
580 REM *** X-BER                                  <127>
589 GOTO 345                                       <249>
590 REM *** Y-BER                                  <201>
599 GOTO 345                                       <003>
600 REM *** Z-BER                                  <019>
601 IF P$="ZEUG"THEN G=6                           <160>
602 IF P$="ZURUECK"THEN V=18                       <011>
```

# Ergänzen Sie jetzt Ihre 64'er-Sammlung

## Schaffen Sie sich ein interessantes Nachschlagewerk und gleichzeitig ein wertvolles Archiv!

Kennen Sie alle Ausgaben von 64'er? Suchen Sie einen ganz bestimmten Testbericht? Oder haben Sie einen Teil eines interessanten Kurses versäumt? Suchen Sie nach einer speziellen Anwendung?

Damit Sie jetzt fehlende Hefte mit »Ihrem« Artikel nachbestellen können, finden Sie auf diesen Seiten eine Zusammenstellung aller wesentlichen Artikel der Ausgaben 01 bis 12/85.

Und so kommen Sie schnell an die noch lieferbaren Ausgaben: Prüfen Sie, welche Ausgabe in Ihrer Sammlung noch fehlt, oder welches Thema Sie interessiert. Tragen Sie die Nummer dieser Ausgabe und das Erscheinungsjahr (z.B. 2/85) auf dem Bestellabschnitt der hier eingehefteten Bestell-Zahlkarte ein. Die ausgefüllte Zahlkarte einfach heraustrennen und Rechnungsbetrag beim nächsten Postamt einzahlen. Ihre Bestellung wird nach Zahlungseingang umgehend zur Auslieferung gebracht.

### Aktuell

| Stichwort | Titel | Seite | Ausgabe |
|---|---|---|---|
| Inhalt | Jahresinhaltsverzeichnis 4/84 bis 3/85 | 84 | 04/85 |
| Allgemeines | Commodore Gestern Heute Morgen | 10 | 01/85 |
| Computer | Amiga — Der neue Supercomputer | 8 | 09/85 |
| DFÜ | MCI Mail — Die schnelle Post | 8 | 02/85 |
| Interview | Interview mit David Crane (Game Designer) | 146 | 06/85 |
| Lernen | Schule braucht Computer (VAM-Computer) | 9 | 06/85 |
| Messen | International Chaos Communication Congress | 15 | 03/85 |
| | Heiße Messe in der Wüste: CES | 8 | 03/85 |
| | Die Sportler kommen (CES Software-Bericht) | 9 | 04/85 |
| | Musikmesse Frankfurt | 8 | 04/85 |
| | Hannover-Messe '85 | 8 | 06/85 |
| | Hannover-Messe '85 | 8 | 07/85 |
| | Chicago im Zeichen der CES | 8 | 08/85 |
| | Aktuelles von der C'85 in Köln | 15 | 08/85 |
| | Btx Total (Internationale Funkausstellung) | 8 | 10/85 |
| | PCW-Computermesse in London | 8 | 11/85 |
| | Neues von der Commodore-Fachausstellung 1985 | 8 | 12/85 |
| Recht | Die neue Abmahnmasche — Vorsicht bei Programmangeboten | 8 | 05/85 |
| | Die Ex-Knacker — wo sind sie geblieben? | 27 | 08/85 |
| | Interview mit Raubkopierern (Section 8) | 28 | 08/85 |
| | Schützer kontra Knacki's | 23 | 08/85 |
| | Raub-Talkshow | 12 | 08/85 |
| | Das Urheberrechtsgesetz und Gedanken zu seiner Anwendung | 21 | 08/85 |
| | Änderung des Urheberrechtsgesetzes | 162 | 09/85 |

### Buchbesprechungen

| Stichwort | Titel | Seite | Ausgabe |
|---|---|---|---|
| Anfänger | Goldmann Computer Compact | 87 | 03/85 |
| | Basic-Wegweiser für den C 64 | 86 | 05/85 |
| | Alles über den C 64, Sachbuchreihe, Band 1 | 115 | 06/85 |
| | Lehrspielzeug Computer: C 64/VC 20 | 112 | 11/85 |
| | C 64 Computerhandbuch | 171 | 11/85 |
| | Einführungskurs: Commodore 64 | 144 | 12/85 |
| Anwendung | Dienstprogramme VC 20, C 64 und SX | 86 | 05/85 |
| | Spaß an Mathe mit dem Commodore 64 | 88 | 07/85 |
| | Mathe für die Oberstufe mit dem C 64 | 88 | 07/85 |
| | Mathematische Routinen VC 20, Elektrotechnik/Elektronik | 112 | 11/85 |
| | Commodore 64-Listings, Band 2: Dateiverwaltung, Schule, Hobby | 112 | 11/85 |
| | Das Trainingsbuch zum Datamat | 144 | 12/85 |
| C 128 | Bücher zum C 128 | 22 | 10/85 |
| DFÜ | Das Mailbox-Jahrbuch: Nutz die Netze | 112 | 11/85 |
| Grafik | Grafik auf dem Commodore 64 (+Fehlert. 9/85) | 86 | 05/85 |
| | Einführung in CAD mit dem Commodore 64 | 128 | 06/85 |
| | Grafik & Musik auf dem Commodore 64 | 88 | 07/85 |
| | Verschiedene Grafikbücher zum C 64 | 115 | 08/85 |
| Programmieren | Von Basic zu Assembler: Das Commodore-Buch, Band 4 | 115 | 06/85 |
| | 64 Intern | 115 | 06/85 |
| | Das Interface Age System-Handbuch zum C 64 | 115 | 06/85 |
| | Das C 64 Buch, Band 5: Simons Basic Leitfaden | 144 | 12/85 |
| | Basicode | 144 | 12/85 |
| | Noch mehr Tips und Tricks zum 64er | 144 | 12/85 |
| Speichern | Das Kassettenbuch zum C 64 und VC 20 | 87 | 03/85 |
| | Die Floppy 1541 (M&T) | 88 | 07/85 |
| Spiele | Rombachs C 64 Spielführer | 87 | 03/85 |
| | Commodore 64-Listings, Band 1, Spiele | 112 | 11/85 |
| | 35 ausgesuchte Spiele für Ihren Commodore 64 | 171 | 1/85 |

### 64'er Extra

| Stichwort | Titel | Seite | Ausgabe |
|---|---|---|---|
| Prozessor | Befehlssatz des 6502/6510-Prozessors | 84 | 09/85 |
| Grafik | Die Videochip-Register des C 64 | 92 | 10/85 |
| Sound | Der SID-Chip, seine Register und Programmierung | 92 | 11/85 |
| Speicher | Die Speicherbelegung des C 64 | 96 | 12/85 |

### Abenteuerlösungen

| Stichwort | Titel | Seite | Ausgabe |
|---|---|---|---|
| Lösungen | Dallas-Quest Lösung | 90 | 01/85 |
| | Lösung The Hobbit | 49 | 02/85 |
| | Guncho Krill-Enchanter ist gelöst | 44 | 03/85 |
| | Infocom-Geheimnisse gelüftet? | 49 | 05/85 |
| | Des Rätsels Lösung: Amazon | 145 | 06/85 |
| | Activision-Adventures entschleiert (Mindshadow, Tracer Sanction) | 36 | 12/85 |
| | Eureka! — ich hab's! | 37 | 12/85 |
| | Lösungen zu Hitchhiker's Guide und Sorcerer | 39 | 12/85 |

### Spiele-Tests

| Stichwort | Titel | Seite | Ausgabe |
|---|---|---|---|
| 007 | James Bond — A View to a Kill | 156 | 09/85 |
| Abenteuer | Abenteuerpaket 1 | 48 | 08/85 |
| | Amazon — das besondere Adventure | 49 | 04/85 |
| | Gordon Saga | 48 | 02/85 |
| | Shadowfire | 146 | 09/85 |
| | The Quest — mit C 64 auf Suche nach Drachen | 47 | 01/85 |
| Action | Hexenküche | 50 | 07/85 |
| | Impossible Mission | 46 | 02/85 |
| | Master of the Lamps | 48 | 07/85 |
| | Rescue on Fractalus | 158 | 10/85 |
| | Stellar 7 | 49 | 08/85 |
| Construction Set | Mail Order Monsters | 49 | 08/85 |
| | Racing Destruction Set | 50 | 08/85 |
| Geschicklichkeit | Australopedicus Robustus | 50 | 08/85 |
| | Boulder Dash II | 159 | 10/85 |
| | Crystal Castles | 50 | 07/85 |
| | Gribbly's Day out | 148 | 09/85 |
| | Rock'n Bolt | 48 | 08/85 |
| | Thing on a Spring | 159 | 10/85 |
| | Tom + Zaga | 48 | 01/85 |
| Pseudo-Adventures | Roland's Rat Race | 49 | 08/85 |
| | Fourth Protocol und Frankie g.t.H. | 162 | 11/85 |
| Renner | Die Renner 1985: Meistverkaufte Spiele | 34 | 12/85 |
| Schach | Viermal Schachmatt: Verschiedene Schachprogramme | 32 | 12/85 |
| Simulation | Elite | 148 | 09/85 |
| | Jump Jet | 148 | 09/85 |
| | Super Huey Hubschraubersimulator | 49 | 07/85 |
| Sport | Boxspiele: Frank Bruno's B. + Barry McGuigan Champions. B. | 49 | 12/85 |
| | Handkantenschlag per Joystick: Karateka + Exploding Fist | 165 | 11/85 |
| | Nick Faldo Plays the Open (Golf) | 159 | 10/85 |
| | Rallye Speedway | 49 | 07/85 |
| | Slapshot (Eishockey) | 50 | 07/85 |
| | Summer Games II | 146 | 09/85 |
| | World Series Baseball | 49 | 07/85 |
| Diverses | New York City und Air Support | 145 | 06/85 |

### Hardware-Tips und Bauanleitungen

| Stichwort | Titel | Seite | Ausgabe |
|---|---|---|---|
| Audio/Video | Besseres Monitorbild beim C 64 | 90 | 02/85 |
| | Richtig verbunden — Video/Audio Ka[illegible] (+ Fehlerteufel 3/85) | 22 | 02/85 |
| | Mit 5 Mark zu neuen Dimensionen (Stereoanlage am C 64) | 34 | 05/85 |
| | Ein Monitor ist genug (RGB+Composite an C 128) | 16 | 10/85 |
| C 16 | Alte Datasette am C 16 | 31 | 04/85 |
| | Alter Joystick am C 16 | 35 | 05/85 |
| Eingabegeräte | Der Hexer — Zusatztastatur für den MSE | 48 | 10/85 |
| EPROM | EPROMs im Expansion-Port | 46 | 10/85 |
| | EPROM-Trans — Die Super-Erweiterung | 42 | 10/85 |
| | Das 64'er EPROM-Programmiergerät, Teil 1 | 44 | 12/85 |
| Floppy/Datasette | Diskettenlaufwerk 1541 selbst justiert | 32 | 10/85 |
| | Die Datasette streikt nie wieder (Anpassung des Tonkopfs) | 34 | 10/85 |
| IEC-Bus | Auf zu neuen Welten: IEC-Bus im Selbstbau (+Fehlerteufel 10/85) | 44 | 07/85 |
| Joystick | Joystick im Selbstbau | 33 | 03/85 |
| | Dauerfeuer-Adapter | 46 | 08/85 |
| RS232/V.24 | Das 30-Mark-Interface (Selbstbau RS232) | 29 | 03/85 |
| | Genau betrachtet: Die RS232/V.24-Schnittstelle | 80 | 05/85 |
| VC 20 | 16 KByte-Erweiterung umschaltbar | 20 | 02/85 |
| | Der VC 20 steuert Super 8-Kamera | 70 | 02/85 |
| Diverses | Userport-Display | 36 | 05/85 |
| | Reset-Taster für alle Fälle (+Fehlert. 9/85) | 130 | 06/85 |
| | Aus eins mach vier (absturzfreie Betriebssystemumschaltung) | 41 | 07/85 |

### Hardware-Grundlagen

| Stichwort | Titel | Seite | Ausgabe |
|---|---|---|---|
| C 16 | C 16 — großer oder kleiner Bruder des C 64? | 29 | 04/85 |
| Computer | Was bringt der C 128? | 28 | 11/85 |
| Drucker | Welcher Drucker ist der Richtige? (Grundlagen) | 15 | 05/85 |
| | Hammerwerke — wie funktionieren Typenraddrucker | 32 | 06/85 |
| | Die Alternativen: Thermo-, Tintenstrahldrucker + Plotter | 24 | 07/85 |
| Eingabegeräte | Versteht Sie Ihr Computer? (Wie funktionieren Eingabegeräte) | 44 | 09/85 |
| Floppy | Floppy oder Datasette? | 129 | 06/85 |
| Monitore | Wie funktionieren sie, was ist beim Kauf zu beachten? | 16 | 12/85 |
| | Das Kabel zum Monitor: Welche Normen gibt es? | 28 | 12/85 |
| Peripherie | Grafikeingabegerät: Wie funktionieren sie? | 30 | 08/85 |

### Hardware-Tests

| Stichwort | Titel | Seite | Ausgabe |
|---|---|---|---|
| 80 Zeichen | Mit 80 fängt das Leben an (Test 80-Zeichen-Karten) (+Fehlert. 6/85) | 17 | 04/85 |
| Computer | Generationswechsel: Test C 16 | 16 | 01/85 |
| | Plus und Minus beim Plus/4 | 14 | 02/85 |
| | PC 128 — der Profi (Hardwaretest) | 13 | 04/85 |
| | Erster ausführlicher Test C 128 PC (Teil 1) | 16 | 06/85 |
| | Erster ausführlicher Test C 128 PC (Teil 2) | 17 | 07/85 |
| DFÜ | Marktübersicht Modems & Akustikkoppler | 32 | 07/85 |
| Drucker | Vergleich: Drucker unter 700 Mark (Tests und Marktübersicht) | 18 | 05/85 |
| | Tests und Marktübersicht Typenraddrucker | 35 | 06/85 |
| | Test: Brother EP 44 | 27 | 07/85 |
| | Brother TC-600 | 118 | 08/85 |
| | Riteman C+ | 133 | 09/85 |
| | Panasonic KX-P1091 | 134 | 09/85 |
| | Star SG 10C | 132 | 09/85 |
| | Melchers CP-80X — wie hätten Sie's denn gern? | 25 | 10/85 |
| | Geheimtip: Der RFI DP 165 | 24 | 10/85 |
| | Epson GX 80 — einer für alle | 26 | 10/85 |
| | MPS 803 — ein Drucker für alle Gelegenheiten? | 40 | 1/85 |
| | Epson JX-80 das vielfarbige Druck-Genie | 38 | 11/85 |
| | Epson FX-85 neue Referenz | 42 | 11/85 |
| | SP 1000 VC — Superstar mit Haken | 41 | 11/85 |
| | Der NEC-P2 — das fernöstliche Wunder | 159 | 12/85 |
| | DMPG9 — eine solide Sache | 162 | 12/85 |
| Eingabegeräte | Der Bildschirm wird zur Leinwand (TechSketch-Lightpen) | 21 | 04/85 |
| | Das Doppelleben des Joystick-Ports: 10er-Tastaturen | 50 | 09/85 |
| | Joysticks: Test und Marktübersicht (+Fehlerteufel 12/85) | 19 | 11/85 |
| | Es geht auch anders: Lightpens und Trackballs | 22 | 11/85 |
| EPROMer | Frisch gebrannt ist halb gespeichert (EPROM-Programmiergeräte im Test) | 39 | 07/85 |
| | QuickByte II — das Kraftpaket | 14 | 10/85 |
| Floppy/Datasette | Schnell wie der Wind (Test: Speeddos, Turbo-Access) | 22 | 04/85 |
| | Turbo-Floppies, zweite Generation: Speeddos plus + Prologic DOS | 28 | 10/85 |
| | Das große Rennen: Schnelle Bandlaufwerke | 37 | 10/85 |
| | Professionelle Floppylaufwerke für den C 64 (IEC-Floppies) | 30 | 10/85 |
| | Gut gekauft ist halb gespeichert (Marktübersicht Disketten) | 38 | 10/85 |
| Grafik | Die Videowerkstatt (Digitizer-Test) | 32 | 05/85 |
| | Digitalbilder m.d. C 64: PrintTechnik Digitizer | 24 | 01/85 |
| Interface | Hardware-Interface ganz weich: Test EC 64 | 23 | 01/85 |
| | Gute Connections — Übersicht Schnittstellen | 21 | 03/85 |
| | Card/Print +6 — Das Allround-Interface | 20 | 03/85 |
| | Das Wiesemann-Centronics-Interface | 18 | 03/85 |
| | Erst ein IEC-Bus öffnet Tür und Tor (+Fehlert.4/6-85) | 24 | 03/85 |
| Monitore | Eine klare Sache: Test Phönix-Monitor | 28 | 04/85 |
| | Marktübersicht: Monochrome Monitore | 30 | 12/85 |
| Musik | Die Stimme des Meisters: Test Voice Master | 19 | 02/85 |
| | Trommelwirbel: Test Digital Drums | 45 | 08/85 |
| | Die Musikhardware zum C 64 | 17 | 09/85 |
| Oszilloskop | Der C 64 als Speicheroszilloskop | 26 | 04/85 |
| Roboter | Roboter selbst gebaut (Fischertechnik) | 167 | 10/85 |
| Scanner | So lernt Ihr Drucker lesen | 30 | 06/85 |
| Speicher | Speichertuning VC 20: Test 64 KByte Karte | 26 | 01/85 |
| Steuern | Flottes Türmchen: MEA-Interface | 116 | 08/85 |

### Kurse

| Stichwort | Titel | Seite | Ausgabe |
|---|---|---|---|
| Assembler | Assembler ist keine Alchimie, Teil 5 | 142 | 01/85 |
| | Assembler ist keine Alchimie, Teil 6 | 134 | 02/85 |
| | Assembler ist keine Alchimie, Teil 7 | 124 | 03/85 |
| | Assembler ist keine Alchimie, Teil 8 | 138 | 04/85 |
| | Assembler ist keine Alchimie, Teil 9 | 138 | 05/85 |
| | Assembler ist keine Alchimie, Teil 10 | 127 | 07/85 |
| | Assembler ist keine Alchimie, Teil 11 | 126 | 08/85 |
| | Assembler ist keine Alchimie, Teil 12 | 109 | 09/85 |
| | Assembler ist keine Alchimie, Teil 13 (Schluß) | 143 | 10/85 |
| C 128 | Entdeckungsreise duch den C 128 | 42 | 12/85 |
| Comal | Comal — Eine Einführung, Teil 3 | 130 | 02/85 |
| Effektives Programmieren | Müllabfuhr im Computer: Garbage Collection, Teil 1 | 122 | 01/85 |
| | Stringprogrammierung in Maschinensprache, Teil 2 | 147 | 02/85 |
| | Finden mit System, eine neuartige Suchmethode, Teil 3 | 148 | 03/85 |
| | Sortieren mit dem Computer, Teil 1 | 148 | 04/85 |
| | Sortieren mit dem Computer, Teil 2 | 159 | 05/85 |
| | Sortieren mit dem Computer, Teil 3 | 124 | 06/85 |
| | Sortieren mit dem Computer, Teil 4 | 138 | 08/85 |
| | Sortieren mit dem Computer, Teil 5 | 124 | 09/85 |
| | Sortieren mit dem Computer, Teil 6 (Schluß) | 150 | 12/85 |
| Extern | C 64 extern — Der Weg nach draußen, Teil 1 | 144 | 08/85 |
| | C 64 extern — Der Weg nach draußen, Teil 2 | 122 | 09/85 |
| | C 64 extern — Der Weg nach draußen, Teil 3 (Schluß) | 129 | 10/85 |
| Floppy | In die Geheimnisse der Floppy eingetaucht, Teil 4 | 148 | 01/85 |
| | In die Geheimnise der Floppy eingetaucht, Teil 5 | 130 | 03/85 |
| | In die Geheimnise der Floppy eingetaucht, Teil 6 | 145 | 05/85 |
| | In die Geheimnisse der Floppy eingetaucht, Teil 7 (Schluß) | 116 | 06/85 |
| | Directory-Manipulationen I | 140 | 06/85 |
| Floppy | Directory-Manipulationen II | 163 | 10/85 |
| Grafik | Hires 3 — die Grafikerweiterung zum Grafikkurs, Teil 1 | 123 | 02/85 |
| | Hires 3 — 15 neue Basic-Befehle, Teil 2 | 136 | 03/85 |
| | Hires 3 — Grafikkurs-Anwendung, Teil 3 (Schluß) | 152 | 08/85 |
| | Sprites ohne Geheimnisse | 40 | 08/85 |
| | Streifzüge durch die Grafikwelt, Teil 1 | 106 | 09/85 |
| | Streifzüge durch die Grafikwelt, Teil 2 | 149 | 11/85 |
| Logeleien | Logeleien, Teil 1 | 143 | 07/85 |
| | Logeleien, Teil 2 | 136 | 08/85 |
| | Logeleien, Teil 3 (Schluß) | 115 | 09/85 |
| Musik | Dem Klang auf der Spur, Teil 2 | 136 | 01/85 |
| | Dem Klang auf der Spur, Teil 3 | 152 | 02/85 |
| | Dem Klang auf der Spur, Teil 4 | 131 | 04/85 |
| | Dem Klang auf der Spur, Teil 5 | 152 | 05/85 |
| | Dem Klang auf der Spur, Teil 7 | 132 | 07/85 |
| | Dem Klang auf der Spur, Teil 8 | 133 | 08/85 |
| | Dem Klang auf der Spur, Teil 9 | 126 | 10/85 |
| | Dem Klang auf der Spur, Teil 10 (Schluß) | 157 | 11/85 |
| Speicher | Memory Map mit Wandervorschlägen, Teil 3 | 126 | 01/85 |
| | Memory Map mit Wandervorschlägen, Teil 4 | 150 | 02/85 |
| | Memory Map mit Wandervorschlägen, Teil 5 | 144 | 03/85 |
| | Memory Map mit Wandervorschlägen, Teil 6 | 144 | 04/85 |
| | Memory Map mit Wandervorschlägen, Teil 7 | 120 | 06/85 |
| | Memory Map mit Wandervorschlägen, Teil 8 | 140 | 07/85 |
| | Memory Map mit Wandervorschlägen, Teil 9 | 129 | 08/85 |
| | Memory Map mit Wandervorschlägen, Teil 10 | 112 | 09/85 |
| | Memory Map mit Wandervorschlägen, Teil 11 | 133 | 10/85 |
| | Memory Map mit Wandervorschlägen, Teil 12 | 145 | 11/85 |

| Stichwort | Titel | Seite | Ausgabe |
|---|---|---|---|
| | Memory Map mit Wandervorschlägen, Teil 13 | 146 | 12/85 |
| Sprachen | Basic ist out — es lebe Forth | 43 | 01/85 |
| VC 20 | Der gläserne VC 20, Teil 4 | 130 | 01/85 |
| | Der gläserne VC 20, Teil 5 | 141 | 02/85 |
| | Der gläserne VC 20, Teil 6 (Schluß) | 155 | 03/85 |

## Software-Tips

| Stichwort | Titel | Seite | Ausgabe |
|---|---|---|---|
| C 128 | Erste Fragen und Antworten zum C 128 | 14 | 09/85 |
| | Fragen und Antworten zum 128er | 20 | 10/85 |
| | Fragen und Antworten zum 128er | 40 | 12/85 |
| Drucker | Der MPS 802 lernt Deutsch | 30 | 05/85 |
| | Centronics-Interface für jeden Bedarf | 78 | 07/85 |
| Textverarbeitung | Software Corner — professionelle Programme richtig eingesetzt (Vizawrite-Tips) | 174 | 12/85 |
| Tips & Tricks | Autoboot beim C 64 | 86 | 03/85 |
| | Verbindungsfreundlich (Parallelschnittstelle des VC 20) | 91 | 03/85 |
| | Undefinierte Opcodes des 6502 | 84 | 03/85 |
| | Durch POKEs zum Erfolg (Spiele-POKEs) | 83 | 03/85 |
| | Tips und Erweiterungen zu Hi-Eddi und Simons Basic | 88 | 03/85 |
| | Hardcopy mit einer Zeile (MPS 801) | 82 | 04/85 |
| | VC 20-Programme schützen | 83 | 04/85 |
| | 64-Tastaturänderung | 63 | 04/85 |
| | Basic-Befehle im Griff | 79 | 05/85 |
| | Durch POKEs zum Erfolg: Spiele-POKEs | 78 | 06/85 |
| | Formatierte Eingabe | 148 | 06/85 |
| | Hi-Text (Text in Hires) | 70 | 08/85 |
| | Verbotene Variablen | 66 | 09/85 |
| | Verschiedene Routinen für Anfänger und Profis (+Fehlerteufel 12/85) | 88 | 11/85 |
| | Der Trick mit dem Joystick (Joystickabfrage) | 24 | 11/85 |
| | Verschiedene Tips für Anfänger und Fortgeschrittene | 106 | 12/85 |

## Software-Grundlagen

| Stichwort | Titel | Seite | Ausgabe |
|---|---|---|---|
| Assembler | Assembler? Assembler! (Einführung) | 32 | 01/85 |
| | Assembler-Bedienung leicht gemacht, Teil 1 | 169 | 12/85 |
| Compiler | So arbeiten Compiler | 39 | 02/85 |
| DFÜ | Ein modernes Abenteuer — Mailboxen in Deutschland | 43 | 04/85 |
| | Der erste Kontakt mit DFÜ | 40 | 06/85 |
| | Die Netze der Post: Btx, Datex-P, Telebox | 46 | 06/85 |
| | DFÜ — Was ist das? | 44 | 06/85 |
| | Mailbox für Anfänger | 30 | 07/85 |
| Datei | Die wichtigsten Begriffe der Dateiverwaltung | 42 | 05/85 |
| | Dateiverwaltung ist nicht gleich Datenbank | 44 | 05/85 |
| | Dateiverwaltung: Was Sie beim Kauf beachten sollten | 40 | 05/85 |
| Drucker | Hardcopy leicht gemacht (wie programmiert man Hardcopies) | 34 | 09/85 |
| EPROM | Wie sage ich es meinem EPROM? (EPROM-Grundlagen) | 35 | 07/85 |
| Funktionen | Funktionen für Anfänger | 164 | 05/85 |
| Lernen | Besser lernen mit dem Computer | 166 | 10/85 |
| Musik | Klangprogrammierung ohne Ballast | 19 | 09/85 |
| Spiele | Taktik- und Strategiespiele | 46 | 03/85 |
| | Play by Mail und Play by Modem | 153 | 09/85 |
| Sprachen | Sprachen für Computer | 47 | 04/85 |
| | Sprachen für Computer, Teil 2 | 46 | 05/85 |
| Textverarbeitung | Von der Schreibmaschine zum Textsystem | 34 | 03/85 |

## Listings zum Abtippen

| Stichwort | Titel | Seite | Ausgabe |
|---|---|---|---|
| Anwendung | Der C 64 als Handballtrainer (AdM) | 52 | 01/85 |
| | Familienplanung (AdM) | 52 | 02/55 |
| | Ligatab — ohne Organisation kein Tor (LdM) | 50 | 03/85 |
| | Gut Ziel mit dem C 64 — Schützenvereinsergebnisse (AdM) | 52 | 03/85 |
| | Weißt du, wieviel Sternlein stehen (Sternkarte) (AdM) (+Fehlert. 6/85) | 52 | 05/85 |
| | Haushaltsbuchführung (AdM) | 52 | 07/85 |
| | Netzwerkanalyse: Ein Programm für Hobbyelektroniker (AdM) | 52 | 08/85 |
| | Prüfungsfragen (AdM) | 52 | 09/85 |
| | Fit in Latein mit dem C 64 (AdM) | 52 | 10/85 |
| | Lyrik-Maschine (AdM) | 52 | 11/85 |
| | Hypra-Platos (LdM) | 50 | 11/85 |
| | Der Chemie-Assistent (AdM) | 52 | 12/85 |
| | SMON Teil 3: Ohne gutes Werkz. geht es nicht | 69 | 01/85 |
| | SMON Teil 4 (+Fehlerteufel 4/85) | 72 | 02/85 |
| | SMON Teil 5 (+Fehlerteufel 5/85) | 64 | 04/85 |
| | Hypra-Ass (LdM) | 51 | 07/85 |
| | Neues vom SMON (+Fehlerteufel 11/85) | 87 | 10/85 |
| | Reassembler zu Hypra-Ass (+Fehlerteufel 12/85) | 97 | 11/85 |
| | Ergänzungen zu Hypra-Ass (bedingte Verzweigungen) | 96 | 11/85 |
| | Tips & Tricks zum SMON (inklusive Diskmonitor) | 100 | 12/85 |
| Basic-Erweiterung | Befehlserweiterung C 64: Bildschirmsteuerung und Masken | 80 | 04/85 |
| | xBasic 64: eine Super-Basic-Erweiterung (LdM) (+Fehlerteufel 5/85) | 52 | 04/85 |
| Bildschirmseite | Auflösung Wettbewerb Bildschirmseite: Drei Top-Programme | 158 | 09/85 |
| DFÜ | Terminalprogramm der Spitzenklasse (+Fehlerteufel 10/85) | 149 | 07/85 |
| Datei | SMU — Der Maskengenerator (LdM) | 50 | 12/85 |
| Drucker | Print-List (formatierte Listings) | 79 | 04/85 |
| | Hi-Eddi-Druckerroutinen | 69 | 06/85 |
| | C 64 Schreiberling — Drucken wie gemalt | 54 | 10/85 |
| | Koalabilder Farbhardcopy auf Epson JX-80 | 39 | 11/85 |
| Einzeiler | Die nächsten 14 aus d. Einzeilerwettbewerb | 157 | 01/85 |
| | 11 neue Einzeiler! (+Fehlerteufel 5/9/85) | 153 | 04/85 |
| Floppy | Hypra-Load mal 4 (+Fehlerteufel 3/85) | 82 | 01/85 |
| | Neues vom Hypra-Load: Hypra-Perfekt | 75 | 04/85 |
| | Diskettenmonitor | 83 | 08/85 |
| | Disk-Designer | 70 | 09/85 |
| | Herzoperation (Hypra-Load + Hypra-Ass + DOS5.1 + Centronics) | 104 | 11/85 |
| Grafik | Vier Pseudo-VICs mit 32 Sprites | 76 | 01/85 |
| | Hi-Eddi: Zeichen- und Malproggramm (LdM) | 50 | 01/85 |
| | Als die Bilder laufen lernten (Pseudo-Scroll) | 88 | 02/85 |
| | Elektrotechnisches Zeichnen mit dem VC 20 | 71 | 03/85 |
| | Supergrafik III (3D-Grafiken mit dem VC 20) | 73 | 04/85 |
| | Funktionen im Netz (3D-Grafik) | 69 | 04/85 |
| | Window 64 — Fenstertechnik für den Commodore | 87 | 04/85 |
| | Mini-Grafik VC 20, Grafikhilfe | 69 | 05/85 |
| | Trickfilm mit dem C 64: Bewegte 3D-Grafik (LdM) (+Fehlerteufel 6/85) | 51 | 05/85 |
| | Kurvenplotten mit Hardcopy auf dem C 16 | 68 | 06/85 |
| | Doppelte Grafikauflösung für C 128 | 33 | 11/85 |
| | Bilder aus einer anderen Dimension (Apfelmännchen) | 80 | 11/85 |
| Intelligenz | VIC — das intelligente Programm (Wettbewerbssieger) | 173 | 05/85 |
| Musik | Sound Machine (+Fehlerteufel 10/85) | 23 | 09/85 |
| | Sound Master (Basic-Erweiterung) | 31 | 09/85 |
| Schach | Schachmeister erweitert | 68 | 04/85 |
| Spiele | Das Grab des Pharaoh (LdM) (+Fehlerteufel 3/85) | 51 | 02/85 |
| | Q - Bert (VC 20) | 78 | 02/85 |
| | Gehirntraining mit Super Memory | 81 | 02/85 |
| | 6510 — Die Suche nach der Prozessor | 70 | 05/85 |
| | Samurai (Strategiespiel) | 72 | 06/85 |
| | Schach dem C 64: Schachprogramm zum Abtippen | 72 | 08/85 |
| | Spielen auf zwei Bildschirmen: Zeichensatzscrolling (LdM) | 51 | 09/85 |
| | Pac-Man unter der Lupe | 76 | 10/85 |
| | Block Out | 84 | 11/85 |
| | Seekrieg per Telefon (Schiffe versenken per Modem) | 82 | 12/85 |
| Spielehilfe | Die Scroll-Maschine — D. Fenster zur Spielewelt (LdM) (+Fehlert. 11/85) | 52 | 06/85 |
| Sprachen | Tiny Forth Compiler (LdM) (+Fehlert. 9/85) | 51 | 08/85 |
| Textverarbeitung | Hypra-Text (LdM) (+Fehlerteufel 11/85) | 50 | 10/85 |
| | Drucksache — Hypra-Text, Teil 2 | 71 | 11/85 |
| Tips & Tricks | Große Buchstaben | 89 | 01/85 |
| | Restore für Unterprogramme | 90 | 01/85 |

| Stichwort | Titel | Seite | Ausgabe |
|---|---|---|---|
| Tips & Tricks | Parameterübergabe an Maschinenspracheprogramme | 88 | 01/85 |
| | Cursorsteuerung leicht gemacht | 86 | 02/85 |
| | Maschinenspracheprogramme auf Disk speichern | 91 | 02/85 |
| | Basic-Zeilen genau betrachtet | 87 | 02/85 |
| | RAM-Floppy | 92 | 02/85 |
| | 22 Read Error — Theorie und Praxis | 41 | 03/85 |
| | Floppy-Lister (+Fehlerteufel 4/85) | 82 | 03/85 |
| | Longscreen beim VC 20 | 83 | 05/85 |
| | C 16: Help und Trace verbessert | 84 | 05/85 |
| | Ordnung ist das halbe Leben (Directory-Sorter) | 77 | 05/85 |
| | Dokumentationshilfe, Cross-Referenz-Liste C 64 (Wettbewerb) | 155 | 06/85 |
| | Prost mit dem C 64: Gerätesteuerung über Userport (+Fehlerteufel 9/85) | 76 | 06/85 |
| | Fenster-Befehle für den C 16 | 84 | 07/85 |
| | Elektronische Merkzettel | 83 | 07/85 |
| | File-Compactor | 82 | 07/85 |
| | REM-Killer (+Fehlerteufel 9/85) | 75 | 07/85 |
| | Basic-Start-Generator | 74 | 07/85 |
| | Komfortable Ein-/Ausgaberoutine | 77 | 07/85 |
| | Bildschirmmasken leicht erstellt | 86 | 08/85 |
| | Der Bitmap-Compander (HiRes-Bilder komprimieren) | 81 | 08/85 |
| | Hypra-Save | 79 | 08/85 |
| | 'Procedure' — oder der C 64 kann lernen | 78 | 08/85 |
| | Aufgewickelt — Listingscrolling für VC 20 | 63 | 09/85 |
| | Programmgenerator für den C 64 | 86 | 10/85 |
| | Cross-Ref optimiert | 83 | 10/85 |
| | Spieletrainer: Spritekill | 86 | 11/85 |
| | Tipp-Utility | 99 | 12/85 |
| | Der EPROM-Automat (wie man Module macht) | 90 | 12/85 |
| | 80-Zeichen-Grafik für den C 128 | 78 | 12/85 |
| | Hyper Screen (Sprites auf dem Bildschirmrand) | 76 | 12/85 |
| Transfer | Der C 64 als PET: PET-Simulator | 87 | 01/85 |
| Unterprogramme | Formatierte Eingabe | 156 | 01/85 |
| Witz | Notlandung (Das lustigste Programm) | 156 | 02/85 |
| | Epson bedruckt Ostereier (AdM) (+Fehlerteufel 5/85) | 50 | 04/85 |

## Software-Tests

| Stichwort | Titel | Seite | Ausgabe |
|---|---|---|---|
| Assembler | Assembler im Test Teil 1 | 34 | 01/85 |
| | Assembler im Test, Teil 2 | 30 | 02/85 |
| Basic-Compiler | Basic-Compiler im Test (+Fehlerteufel 5/85) | 34 | 02/85 |
| Basic-Erweiterung | GBasic — Alles drin | 28 | 01/85 |
| | Aztec Basic — von jedem etwas | 42 | 04/85 |
| | Macro-Basic: Die Unterprogramm-Bibliothek | 137 | 06/85 |
| | Darf es etwas mehr sein? — Test Business-Basic | 120 | 08/85 |
| | Das Intellectool | 138 | 09/85 |
| | Formel 64: Das Multitalent | 158 | 12/85 |
| Compiler | Basic 64 — ein vielseitiger Basiccompiler | 36 | 04/85 |
| DFÜ | Terminal 64 — Schwer auf Draht | 24 | 02/85 |
| | Terminalprogramme: Übersicht | 42 | 06/85 |
| Datei | Vergleichstest — 7 Dateiverwaltungen auf einen Blick | 118 | 07/85 |
| | Aufgeräumt mit Mainfile II | 157 | 10/85 |
| Grafik | Ich glaub, mein Drucker pfeift (Test: Printshop) | 34 | 04/85 |
| | Malkasten ade! (Test: Blazing Paddles) | 40 | 04/85 |
| | Malen auf dem Bildschirm (Malprogramme) | 34 | 08/85 |
| | Grafikprogramme auf einen Blick: Marktübersicht | 38 | 08/85 |
| | Vergleichstest: Grafik-Erweiterungen | 37 | 09/85 |
| Lernen | Softlearning — die weiche Welle des Lernens | 40 | 01/85 |
| | Nachhilfe (Übersicht Lernsoftware) | 26 | 02/85 |
| | Vokabeltraining mit dem Computer | 39 | 03/85 |
| | Marktübersicht: Lernsoftware | 168 | 10/85 |
| Musik | Musik für den C 64: Übersicht Musiksoftware | 26 | 09/85 |
| | The Music System — Zwei auf einen Schlag | 164 | 12/85 |
| Sprachen | Logo — die Sprache für Einsteiger | 135 | 05/85 |
| | Der Ada Trainingskurs auf dem C 64 | 129 | 05/85 |
| | Promal — die neue Sprache für Profis? | 124 | 07/85 |
| | Forth-wärts mit M&T-Forth 64 | 126 | 07/85 |
| | Was leistet Pilot? | 121 | 08/85 |
| | Pascal für Profis (Profi-Pascal) | 122 | 08/85 |
| | Super-Forth 64 | 144 | 09/85 |
| | C — die professionelle Programmiersprache für den C 64 | 140 | 09/85 |
| | Basic 7.0 — Das Superbasic des C 128 | 18 | 10/85 |
| | Comal 80 — die universelle Programmiersprache | 151 | 10/85 |
| | Turbo-Pascal auf dem C 128 | 30 | 11/85 |
| Textverarbeitung | Homeword - Textverarbeitung zu Hause | 36 | 03/85 |
| | Totl-Text — Flexibilität ist Trumpf | 38 | 03/85 |
| | Texte gut im Griff (Übersicht Textverarbeitung) (+Fehlerteufel 5/85) | 38 | 04/85 |
| | Protext — Textprofi mit 80 Zeichen | 133 | 05/85 |
| | Textomat Plus kontra Vizawrite | 132 | 06/85 |
| | Der Preishammer (Test: StarTexter) | 135 | 09/85 |
| | Paperclip — ausdrücklich gut | 44 | 11/85 |

## So machen's andere

| Stichwort | Titel | Seite | Ausgabe |
|---|---|---|---|
| Lernen | Gelungener Einstieg (Informatik-Unterricht) | 159 | 04/85 |
| Semmeln | Semmelservice mit dem C 64 | 147 | 06/85 |
| Sport | Commodore Sportservice: Heimcomputer zur Turnierauswertung | 157 | 07/85 |
| Hilfe | Computer für Behinderte | 182 | 12/85 |

```
603 IF P$="ZEICHNUNG"THEN S=25                      <098>
604 IF P$="ZUENDSCHNUR"THEN S=28                    <094>
605 IF P$="ZUENDE"THEN V=25                         <046>
609 GOTO 345                                        <013>
700 REM *** STANDARD-ANTWORTEN ***********
    ************************************** <208>
701 PRINT"{DOWN}KEIN WEG IN DIESER RICHTUN
    G !":Q=1:RETURN                                 <069>
702 ZV=RND(1):IF ZV<0.8 THEN PRINT"{DOWN}D
    AS VERSTEHE ICH NICHT!":Q=1:RETURN              <071>
703 PRINT"{DOWN}WAS SOLL DEN{2SPACE}D A S{
    2SPACE}BEDEUTEN ? !":Q=1:RETURN                 <179>
704 PRINT"{UP}BITTE EINE KORREKTE EINGABE!
    ":R=0:Q=1:GOTO 347                              <069>
705 PRINT"{WHITE,DOWN}OK":Q=1:RETURN                <147>
706 PRINT"{DOWN}HIER GIBT ES NICHTS MEHR Z
    U NEHMEN!":Q=1:GOTO 989                         <175>
707 PRINT"{DOWN}SIE HABEN DOCH NICHTS MEHR
    !":Q=1:GOTO 989                                 <176>
708 PRINT"{DOWN}SIE ENTDECKEN NICHTS BESON
    DERES.":Q=1:RETURN                              <132>
709 PRINT"{DOWN}IST SCHON OFFEN!":Q=1:RETU
    RN                                              <075>
710 PRINT"{DOWN}IST SCHON ZU!":Q=1:RETURN           <110>
711 PRINT"{DOWN}SIE WARTEN.DIE ZEIT VERGEH
    T...":Q=1:RETURN                                <074>
712 PRINT"{DOWN}SOVIELE SACHEN KOENNEN SIE
     NICHT TRAGEN!{2UP}":Q=1:RETURN                 <086>
713 PRINT"{UP}WIESO DRUECKEN SIE 'RETURN'
    OHNE ETWAS E{DOWN}INZUGEBEN ?":UA=0:GO
    TO 100                                          <230>
730 R=0:S=0:V=0:G=0:Q=1:RETURN                      <180>
749 REM *** STANDARD-REAKTIONEN **********
    ************************************** <002>
750 GET B$:IF B$<>CHR$(32)THEN 750                  <023>
751 X$="":RETURN                                    <023>
760 FOR I=0 TO 10:IF RA$(RA,I)<>""THEN PRI
    NT"{DOWN,YELLOW}"RA$(RA,I)                      <197>
761 NEXT:RETURN                                     <192>
777 ZV=RND(1):IF Q=0 AND ZV<.7 THEN PRINT"
    {DOWN}DAS GEHT NICHT!"                          <197>
778 IF Q=0 AND ZV>.7 THEN PRINT"{DOWN}WAS
    WOLLEN SIE DENN DAMIT BEZWECKEN?"               <136>
779 RETURN                                          <075>
790 PRINT"{DOWN}DIE KANNIBALEN DROHEN SIE
    EINZUHOLEN!!                                    <053>
791 LI=1200:D=TI:DE=1:RETURN                        <028>
792 PRINT"{DOWN}SIE HOEREN DEN KNALL EINER
     EXPLOSION!":ZA=0:T=0:ZU=2:RA$(14,2)="
    "                                               <142>
793 IF KA=1 THEN PRINT"{DOWN}SIE HABEN DIE
     KANNIBALEN IN DIE FLUCHT{2SPACE}GESCL
    AGEN."                                          <139>
794 RA$(13,2)="":RA$(15,2)="":RA$(20,2)=""
    :RA$(12,2)="":RETURN                            <245>
800 IF T=1 THEN 870                                 <224>
801 IF RA$(12,9)<>""THEN WL=1                       <016>
802 IF RA$(12,9)=""THEN WL=3                        <245>
805 IF V=1 AND WI>WL THEN GOSUB 712:GOTO 8
    38                                              <180>
810 IF V=1 AND RA$(RA,G)<>""THEN GOSUB 705
    :SP$(G)=G$(G):RA$(RA,G)="":WI=WI+1:GOT
    O 849                                           <234>
815 IF V=1 AND G>0 AND SP$(G)=G$(G)THEN PR
    INT"{DOWN}DAS HABEN SIE SCHON!":Q=1:GO
    TO 899                                          <045>
820 IF V=1 AND G>0 AND RA$(RA,G)=""THEN PR
    INT"{DOWN}HIER GIBT ES DAS NICHT!":Q=1
    :GOTO 899                                       <211>
825 IF V=2 AND G>0 AND SP$(G)=""THEN PRINT
    "{DOWN}DAS HABEN SIE DOCH GAR NICHT!":
    Q=1:GOTO 899                                    <052>
830 IF V=2 AND SP$(G)=G$(G)THEN GOSUB 705:
    RA$(RA,G)=G$(G):SP$(G)="":WI=WI-1:GOTO
     849                                            <050>
832 IF V=1 AND S=1 THEN GOSUB 980:Q=1:IF A
    =1 THEN GOSUB 705                               <205>
835 IF V=2 AND S=1 THEN GOSUB 986:Q=1:IF A
    =1 THEN GOSUB 705                               <113>
838 IF S=4 THEN GOSUB 990:Q=1                       <103>
840 IF V=4 THEN GOSUB 711:GOTO 899                  <069>
842 IF V=5 AND SP$(5)<>""THEN GOTO 999              <021>
843 IF V=5 AND SP$(5)=""THEN PRINT"{DOWN}S
    IE HABEN NICHTS ZU ESSEN!":Q=1:GOTO 89
    9                                               <166>
844 IF V=6 AND S>OG OR V=6 AND S<UG THEN G
    OSUB 708                                        <088>
845 IF V=8 AND S=15 AND SP$(3)=G$(3)AND RA
    <>3 THEN GOSUB 3600:Q=1                         <000>
846 IF V=7 AND SP$(2)=G$(2)AND G=2 AND RA<
    >3 THEN GOSUB 3800:Q=1                          <121>
847 IF V=19 THEN PRINT"{DOWN}SIE SINGEN EI
    N LIED...":Q=1                                  <202>
848 IF V=6 AND S=11 AND SP$(5)<>""THEN PRI
    NT"{DOWN}SIE STELLEN FEST,DASS SIE GIF
    TIG SIND!":Q=1                                  <024>
849 IF SP$(9)<>""THEN WL=1                          <218>
850 IF SP$(9)=""THEN WL=3                           <121>
869 RETURN                                          <165>
870 GOSUB 730:IF DE=1 THEN RA=15:GOTO 900           <233>
871 IF KA=1 THEN GOSUB 790                          <222>
872 IF ZU=1 AND SP$(2)<>""THEN RA=12:GOTO
    900                                             <038>
873 IF ZU=1 AND SP$(2)=""THEN GOSUB 792             <063>
899 RETURN                                          <195>
900 REM *** TODESANZEIGEN ****************
    ************************************** <254>
901 GOSUB 30000                                     <135>
902 ON RA GOTO 905,907,909,911,913,915,917
    ,919,922,923,925,927,929,931,933,935,9
    37                                              <022>
903 ON RA-17 GOTO 939,941,943,945,947,949           <067>
905 PRINT"SIE SIND VON DER PALME GESTUERZT
     UND                                            <061>
906 PRINT"HABEN SICH DABEI DAS GENICK GEBR
    OCHEN!":GOTO 30100                              <006>
907 PRINT"DIE FLEISCHFRESSENDE PLANZE HAT
    SIE                                             <102>
908 PRINT"VERSPEIST!":GOTO 30100                    <059>
909 PRINT"DER SKORPION HAT SIE GESTOCHEN!D
    AS GIFT                                         <213>
910 PRINT"HAT SIE GETOETET!!":GOTO 30100            <233>
911 PRINT"DER GORILLA HAT IHNEN DAS GENICK
     GE-                                            <006>
912 PRINT"BROCHEN!":GOTO 30100                      <058>
913 PRINT"SIE SIND IN DEN SUMPF GESTUERZT
    UND                                             <024>
914 PRINT"ERSTICKT!":GOTO 30100                     <204>
915 PRINT"DIE FRUECHTE WAREN GIFTIG!":GOTO
     30100                                          <095>
917 PRINT"DIE GROSSE,SCHWARZE SPINNE HAT S
    IE                                              <243>
918 PRINT"GEBISSEN!":GOTO 30100                     <102>
919 PRINT"DER STEINHAGEL HAT SIE IN DIE TI
    EFE                                             <243>
920 PRINT"GERISSEN!":GOTO 30100                     <108>
922 PRINT"SIE SIND IN DIE TIEFE GESTUERZT!
    !":GOTO 30100                                   <192>
923 PRINT"DIE WAENDE HABEN SIE ZERQUETSCHT
    !!":GOTO 30100                                  <127>
925 PRINT"DIE EINSTUERZENDE TREPPE HAT SIE
     UNTER                                          <105>
926 PRINT"SICH BEGRABEN!":GOTO 30100                <034>
927 PRINT"DER EXPLODIERENDE BEHAELTER HAT
    SIE IN                                          <138>
928 PRINT"STUECKE GERISSEN!":GOTO 30100             <234>
929 PRINT"DAS GIFT DER SCHLANGE HAT IHRE A
    TMUNG{DOWN}                                     <231>
930 PRINT"GELAEHMT.SIE SIND ERSTICKT!":GOT
    O 30100                                         <215>
931 PRINT"DER JAGUAR HAT SIE GETOETET!":GO
    TO 30100                                        <145>
933 PRINT"DIE KANNIBALEN HABEN SIE MIT EIN
    ER KEULE                                        <219>
934 PRINT"ERSCHLAGEN!":GOTO 30100                   <138>
935 PRINT"{6RIGHT}SIE SIND VERBLUTET!":GOT
    O 30100 READ Y.                                 <118>
980 A=0:FOR I=0 TO 10                               <242>
981 :   IF RA$(12,9)<>""THEN WL=1                   <157>
982 :   IF WI>WL THEN GOSUB 712:RETURN              <141>
983 :   IF RA$(RA,I)<>""THEN SP$(I)=RA$(RA
    ,I):RA$(RA,I)="":A=1:WI=WI+1                    <134>
984 NEXT:IF A=0 THEN GOTO 706                       <215>
985 RETURN                                          <025>
986 A=0:FOR I=0 TO 10                               <248>
987 :   IF SP$(I)<>""THEN RA$(RA,I)=SP$(I)
    :SP$(I)="":A=1:WI=WI-1                          <039>
988 NEXT:IF A=0 THEN GOTO 707                       <235>
989 RETURN                                          <029>
990 A=0:PRINT"{DOWN}SIE HABEN:":FOR I=0 TO
     10:IF SP$(I)<>""THEN PRINT"{YELLOW,DO
    WN}"SP$(I):A=1                                  <077>
991 NEXT:IF A=0 THEN PRINT"{YELLOW,DOWN}NI
```

**Listing »Odyssey« (Fortsetzung)**

```
     CHTS"                                   <213>
 992 RETURN                                  <032>
 999 PRINT"{WHITE,DOWN}SIE ESSEN VON DEN FR
     UECHTEN!":FOR I=0 TO 3000:NEXT:RA=6:GO
     TO 900                                  <212>
1000 PRINT SPC(9)"{GREEN,SPACE,GREY 1}GGGG
     GGGGGGGGGGGGGGGG "                      <247>
1001 PRINT SPC(9)"H{GREEN,5SPACE}I{14SPACE
     ,GREY 1}G"                              <251>
1002 PRINT SPC(9)"H{GREEN,2SPACE}G{RVSON}I
     T{RVOFF}TU{RVSON}E{RVOFF}TTU{9SPACE,G
     REY 1}G"                                <214>
1003 PRINT SPC(9)"H{GREEN,SPACE,RVSON}EU I
     TEUG UT{RVOFF,8SPACE,GREY 1}G"          <076>
1004 PRINT SPC(9)"H{GREEN}G{RVSON}EI{RVOFF
     }EU{BROWN,RVSON}TT{GREEN,RVOFF}TIE{RV
     SON}IJ{RVOFF,8SPACE,GREY 1}G"           <100>
1005 PRINT SPC(9)"H{GREEN}U{RVSON}J{RVOFF}
     J{2SPACE,BROWN,RVSON}TT{RVOFF,SPACE,G
     REEN,SPACE}U{RVSON}U{RVOFF}V{8SPACE,G
     REY 1}G"                                <028>
1006 PRINT SPC(9)"H{GREEN,5SPACE,BROWN,RVS
     ON}TT{GREEN,RVOFF,9SPACE}I{3SPACE,GRE
     Y 1}G"                                  <057>
1007 PRINT SPC(9)"H{GREEN,5SPACE,BROWN,RVS
     ON}TT{GREEN,RVOFF,SPACE,BROWN,5SPACE,
     GREEN,RVSON}EUT{RVOFF}E{RVSON}EIT{GRE
     Y 1,RVOFF}G"                            <104>
1008 PRINT SPC(9)"H{BROWN,5SPACE,RVSON}TT{
     RVOFF,5SPACE,GREEN,SPACE,RVSON}UU{RVO
     FF}U{BROWN,RVSON}T{GREEN,RVOFF}JP{RVS
     ON}K{GREY 1,RVOFF}G"                    <247>
1009 PRINT SPC(9)"H{GREEN,5SPACE,BROWN,RVS
     ON}TT{GREEN,RVOFF,6SPACE}CY{SPACE,BRO
     WN,RVSON}T{GREEN,RVOFF,2SPACE}K{GREY
     1}G"                                    <018>
1010 PRINT SPC(9)"H{GREEN,5SPACE,BROWN,RVS
     ON}TT{GREEN,RVOFF,9SPACE,BROWN,RVSON}
     T{GREEN,RVOFF,3SPACE,GREY 1}G"          <091>
1011 PRINT SPC(9)"H{SPACE,GREEN,4SPACE,BRO
     WN,RVSON}TT{GREEN,RVOFF,4SPACE,ORANGE
     ,SPACE,GREEN,4SPACE,BROWN,RVSON}T{GRE
     EN,RVOFF,3SPACE,GREY 1}G"               <016>
1012 PRINT SPC(9)"H{GREEN,4SPACE,BROWN,SPA
     CE,RVSON}TT{GREEN,RVOFF,9SPACE,BROWN,
     RVSON}T{GREEN,RVOFF,3SPACE,GREY 1}G"    <093>
1013 PRINT SPC(9)"H{ORANGE}T{RVSON}UYT{4SP
     ACE}TYU{RVOFF}TOPG{SPACE,BROWN,RVSON}
     T{ORANGE,RVOFF,3SPACE,GREY 1}G"         <173>
1014 PRINT SPC(9)"H{ORANGE,RVSON,SPACE}U{2
     SPACE}U{2SPACE}K{3SPACE}K{6SPACE}TY{G
     REY 1,RVOFF}G"                          <175>
1015 PRINT SPC(9)"H{ORANGE,RVSON,3SPACE}U
     J{3SPACE}U{5SPACE}J{2SPACE}I{SPACE,GR
     EY 1,RVOFF}G"                           <085>
1016 PRINT SPC(9)"{ORANGE,SPACE,GREY 1}TTT
     TTTTTTTTTTTTTTTT "                      <152>
1019 RA=1:OG=6:UG=6                          <051>
1020 PRINT"{GREEN}SIE BEFINDEN SICH AM SUE
     DSTRAND EINER{3SPACE}UNBEKANNTEN INSE
     L.                                      <210>
1050 GOSUB 760                               <122>
1120 PRINT"{WHITE,DOWN}MOEGLICHE RICHTUNGE
     N : N"                                  <127>
1125 GOSUB 750                               <181>
1130 GOSUB 100                               <058>
1135 IF S=5 THEN GOTO 1000                   <143>
1140 ON R GOTO 2000,1160,1160,1160,1600,11
     60                                      <188>
1150 GOTO 1300                               <122>
1160 GOSUB 701                               <168>
1300 IF V=6 AND S=6 THEN PRINT"{DOWN}SIE U
     NTERSUCHEN DEN STRAND.":Q=1:IF R1=0 T
     HEN GOSUB 1500                          <225>
1399 GOSUB 777                               <201>
1400 GOTO 1130                               <182>
1500 IF R1=0 THEN RA$(1,1)=G$(1):R1=1        <101>
1510 GOSUB 760:RETURN                        <029>
1600 PRINT"{DOWN}SIE SITZEN AUF DER PALME.
     "                                       <213>
1610 GOSUB 100                               <030>
1620 ON R GOTO 900,900,900,900,900,1650      <005>
1630 IF V=4 THEN GOTO 1610                   <138>
1640 GOTO 900                                <108>
1650 GOSUB 705:GOTO 1130                     <072>
2000 PRINT SPC(9)"{GREY 1,SPACE}GGGGGGGGGG
     GGGGGGGGGG "                            <068>
2001 PRINT SPC(9)"H{RVSON,SPACE}G{SPACE,GR
     EEN}H{GREY 1,SPACE,RVOFF,SPACE,RVSON}
     T{2SPACE,GREEN}H{GREY 1,SPACE}TG B{SP
     ACE,GREEN}H{GREY 1,3SPACE,RVOFF}G"      <192>
2002 PRINT SPC(9)"H{RVSON,SPACE}G{SPACE,GR
     EEN}N{GREY 1,SPACE,RVOFF}G{RVSON}G H{
     GREEN}N{GREY 1,SPACE}H{2SPACE}B{SPACE
     ,GREEN}T{GREY 1,3SPACE,RVOFF}G"         <198>
2003 PRINT SPC(9)"H{RVSON,SPACE}B{SPACE,GR
     EEN}T{GREY 1,SPACE,RVOFF}H{RVSON,2SPA
     CE}N{GREEN}T{GREY 1,SPACE}B{2SPACE}H{
     SPACE,GREEN,RVOFF}T{GREY 1,RVSON}G{2S
     PACE,RVOFF}G"                           <102>
2004 PRINT SPC(9)"H{RVSON,SPACE}H{SPACE,GR
     EEN,RVOFF}T{GREY 1,RVSON}G{RVOFF}T{RV
     SON}G{2SPACE,GREEN,RVOFF}T{GREY 1,RVS
     ON}H{2SPACE}B{2SPACE,GREEN,RVOFF}T{GR
     EY 1,RVSON,3SPACE,RVOFF}G"              <133>
2005 PRINT SPC(9)"H{RVSON,SPACE}H{SPACE,GR
     EEN,RVOFF}T{GREY 1,RVSON,SPACE,RVOFF}
     T{RVSON,3SPACE,GREEN,RVOFF}T{GREY 1,R
     VSON}HH{4SPACE,GREEN,RVOFF}T{GREY 1,R
     VSON}G{2SPACE,RVOFF}G"                  <046>
2006 PRINT SPC(9)"H{RVSON,SPACE}H{SPACE,GR
     EEN,RVOFF}T{GREY 1,RVSON,SPACE}T{3SPA
     CE,GREEN,RVOFF}T{GREY 1,RVSON}B G{3SP
     ACE,GREEN,RVOFF}G{GREY 1,RVSON,2SPACE
     }N{RVOFF}G"                             <226>
2007 PRINT SPC(9)"H{RVSON,SPACE}H{SPACE,GR
     EEN,RVOFF}H{GREY 1,RVSON,SPACE}NG{2SP
     ACE,GREEN,RVOFF}T{GREY 1,RVSON}G H{3S
     PACE}B{2SPACE}B{RVOFF}G"                <037>
2008 PRINT SPC(9)"H{RVSON,2SPACE}GB H{2SPA
     CE}B{GREEN,RVOFF}T{GREY 1,RVSON,3SPAC
     E}B{5SPACE}H{RVOFF}G"                   <238>
2009 PRINT SPC(9)"H{RVSON,2SPACE}BG{3SPACE
     }HT{GREEN,RVOFF}T{GREY 1,RVSON,3SPACE
     }H{4SPACE}NG{RVOFF}G"                   <023>
2010 PRINT SPC(9)"H{RVSON,2SPACE}B{4SPACE}
     GH{GREEN,RVOFF}T{GREY 1,RVSON}H{2SPAC
     E}HG{3SPACE,RVOFF}T{RVSON}G{RVOFF}G"    <201>
2011 PRINT SPC(9)"H{RVSON,2SPACE}H{3SPACE}
     H{2SPACE,GREEN,RVOFF}H{GREY 1,RVSON,4
     SPACE}G{3SPACE}T{SPACE,RVOFF}G"         <032>
2012 PRINT SPC(9)"H{RVSON,2SPACE}H{3SPACE}
     Y{2SPACE}B{4SPACE}I{3SPACE,RVOFF}T{RV
     SON,SPACE,RVOFF}G"                      <019>
2013 PRINT SPC(9)"H{RVSON,2SPACE}HH{2SPACE
     }H{2SPACE}G{4SPACE}B{3SPACE}Y{SPACE,R
     VOFF}G"                                 <012>
2014 PRINT SPC(9)"H{RVSON,2SPACE}HT{2SPACE
     }B{2SPACE}G{4SPACE}H{3SPACE}B{SPACE,R
     VOFF}G"                                 <060>
2015 PRINT SPC(9)"H{RVSON,2SPACE}TT{2SPACE
     }B{2SPACE}B{4SPACE}Y{3SPACE}G{SPACE,R
     VOFF}G"                                 <224>
2016 PRINT SPC(9)" TTTTTTTTTTTTTTTTTTTT "    <027>
2019 RA=2:OG=1:UG=1                          <172>
2020 PRINT"{GREEN}SIE STEHEN AN EINER HOHE
     H,MIT LIANEN BE-WACHSENEN FELSWAND      <219>
2030 PRINT"{UP,19RIGHT}AN DEREN FUSS EINE    <183>
2040 PRINT"GROSSE,FLEISCHFRESSENDE PLANZE
     WAECHST.                                <051>
2120 PRINT"{WHITE,DOWN}MOEGLICHE RICHTUNGE
     N : S,W,O"                              <177>
2125 GOSUB 750                               <165>
2130 GOSUB 100                               <042>
2135 IF S=5 THEN GOTO 2000                   <128>
2140 ON R GOTO 2160,1000,5000,2500,2200,21
     60                                      <189>
2150 GOTO 2165                               <083>
2160 GOSUB 701                               <152>
2165 GOSUB 777                               <205>
2170 GOTO 2130                               <198>
2200 PRINT"{DOWN,GREEN}SIE SIND AN EINER L
     IANE HINAUFGEKLETTERT.                  <011>
2222 PRINT"{UP,RIGHT}DA DIESE IN HALBER HO
     EHE RISS,SIND SIE"                      <022>
2224 PRINT"AUF DIE ERDE ZURUECKGEFALLEN!     <197>
2226 PRINT"AUSGERECHNET IN DEN SCHLUND DER
      FLEISCH-FRESSENDEN PLANZE!!            <209>
2228 PRINT"JETZT WERDEN SIE VERDAUT!!!       <186>
2300 GOSUB 750:GOTO 900                      <255>
2500 REM *** RAUM 2500 *******************
     **********************************
```

```
     *                                            <094>
2510 PRINT"{GREEN,DOWN}SIE STEHEN AUF DEM
     WESTLICHEN VORPLATZ{2SPACE}EINER VERL
     ASSENEN HUETTE.                              <203>
2520 PRINT"AM BODEN SIND ABDRUECKE NACKTER
      FUESSE{2SPACE}UND SCHLEIFSPUREN ZU E
     RKENNEN                                      <121>
2530 PRINT"{11LEFT}."                             <146>
2540 PRINT"HIER SCHEINT EIN KAMPF STATTGEF
     UNDEN ZU HABEN!                              <148>
2545 RA=20:OG=1:UG=1                              <200>
2550 GOSUB 760                                    <096>
2600 PRINT"{WHITE,DOWN}MOEGLICHE RICHTUNGE
     N : S,W,O{DOWN}"                             <046>
2610 GOSUB 100                                    <014>
2620 ON R GOTO 2640,4000,2000,3000,2640,26
     40                                           <141>
2625 IF S=5 THEN GOTO 2500                        <120>
2630 GOTO 2645                                    <071>
2640 GOSUB 701                                    <124>
2645 GOSUB 777                                    <177>
2650 GOTO 2610                                    <186>
3000 PRINT SPC(9)"{BROWN,SPACE,GREY 1}@@@@
     @@@@@@@@@@@@@@@@{BROWN,SPACE}"               <136>
3001 PRINT SPC(9)"{GREY 1}H{BROWN,RVSON}H{
     SPACE,GREEN}H{BROWN}HHHHHHHHHHHHHHHHB
     {GREY 1,RVOFF}G"                             <219>
3002 PRINT SPC(9)"H{BROWN,RVSON}HH{GREEN}H
     {BROWN}HHHH{GREEN}HHE+S{BROWN}HHHHHHH
     B{GREY 1,RVOFF}G"                            <238>
3003 PRINT SPC(9)"H{BROWN,RVSON}HL{GREEN}L
     {BROWN}HHHH{GREEN}BGGFB{BROWN}HHHHHHH
     B{GREY 1,RVOFF}G"                            <165>
3004 PRINT SPC(9)"H{BROWN,RVSON}NH{GREEN,R
     VOFF}T{BROWN,RVSON}HHHH{GREEN}GG+XW{B
     ROWN}HHHHHHHB{GREY 1,RVOFF}G"                <204>
3005 PRINT SPC(9)"H{BROWN,RVSON}HW{GREEN,R
     VOFF}T{BROWN,RVSON}HHHH{GREEN}B.BOB{B
     ROWN}HHHHHHHB{GREY 1,RVOFF}G"                <090>
3006 PRINT SPC(9)"H{BROWN,RVSON}H{SPACE,GR
     EEN,RVOFF}H{BROWN,RVSON}HHHH{GREEN}Z*
     E*X{BROWN}HHHHHHHB{GREY 1,RVOFF}G"           <202>
3007 PRINT SPC(9)"H{BROWN,RVSON}H GHHHHHHH
     HHHHHHHHHB{GREY 1,RVOFF}G"                   <225>
3008 PRINT SPC(9)"H{BROWN,RVSON}H{SPACE,RV
     OFF}£{GREY 1}HS{SPACE,BROWN,SPACE,WHI
     TE,RVSON}£V{RVOFF}£{BROWN,RVSON}£HHBG
     GHHHB{GREY 1,RVOFF}G"                        <141>
3009 PRINT SPC(9)"H{BROWN,RVSON}B{RVOFF}£{
     SPACE,GREY 1}JK{SPACE,WHITE,3SPACE,BR
     OWN,RVSON}£GGGYULGHB{GREY 1,RVOFF}G"         <071>
3010 PRINT SPC(9)"H{BROWN,RVSON,3SPACE}JH{
     3SPACE}JH UYG GIUMB{GREY 1,RVOFF}G"          <069>
3011 PRINT SPC(9)"H{BROWN,RVSON,SPACE}U JH
      U JH{2SPACE}M{SPACE,WHITE}X{BROWN,SP
     ACE}G{2SPACE}M{GREY 1,RVOFF}G"               <061>
3012 PRINT SPC(9)"H{BROWN,RVSON,3SPACE}T I
      {2SPACE}T U YG GTI{2SPACE,GREY 1,RVOF
     F}G"                                         <215>
3013 PRINT SPC(9)"H{BROWN,RVSON}I{2SPACE}T
      {4SPACE}T{3SPACE}HGTGH{3SPACE,GREY 1,
     RVOFF}G"                                     <252>
3014 PRINT SPC(9)"H{BROWN,RVSON,4SPACE}U{6
     SPACE}I{2SPACE}T{2SPACE}U{2SPACE,GREY
      1,RVOFF}H"                                  <002>
3015 PRINT SPC(9)"{BROWN,SPACE,GREY 1}TTTT
     TTTTTTTTTTTTTTTT{BROWN,SPACE}"               <119>
3019 RA=3:OG=10:UG=8                              <154>
3020 PRINT"{GREEN}DIE HUETTE SCHEINT VOR K
     URZEM NOCH BEWOHNT GEWESEN ZU SEIN.          <099>
3022 PRINT"AUF DEM BODEN STEHT EIN FASS.          <207>
3024 GOSUB 760                                    <062>
3120 PRINT"{WHITE,DOWN}MOEGLICHE RICHTUNGE
     N : W"                                       <131>
3125 GOSUB 750                                    <149>
3130 GOSUB 100                                    <026>
3135 IF S=5 THEN GOTO 3000                        <113>
3140 ON R GOTO 3160,3160,2500,3160,3160,31
     60                                           <064>
3150 GOTO 3300                                    <106>
3160 GOSUB 701                                    <136>
3300 IF V=6 AND S=8 THEN PRINT"{DOWN}ES EN
     THAELT SCHIESSPULVER.":Q=1                   <015>
3310 IF V=6 AND S=9 AND SP$(2)=""AND SP$(3
     )=""THEN GOSUB 3500                          <254>
3320 IF V=6 AND S=10 AND R3=1 THEN PRINT"{
     DOWN}ES IST EIN TAGEBUCH.":Q=1               <056>
3350 IF V=8 AND S=10 AND R3=1 THEN GOSUB 3
     600                                          <062>
3360 IF V=7 AND S=8 THEN GOSUB 3700               <130>
3375 IF V=7 AND G=2 AND R3=1 THEN GOSUB 38
     00                                           <045>
3380 IF V=11 AND S=21 AND RA=3 AND OF=1 AN
     D OB=1 AND G=2 THEN GOSUB 705:GOSUB 3
     900:PA=1                                     <093>
3385 IF V=10 AND S=8 AND OF=0 THEN GOSUB 7
     10                                           <038>
3387 IF V=10 AND G=2 AND OB=0 THEN GOSUB 7
     10                                           <132>
3390 IF V=10 AND S=8 AND OF=1 THEN GOSUB 7
     05:OF=0                                      <066>
3391 IF V=10 AND G=2 AND OB=1 THEN GOSUB 7
     05:OB=0                                      <157>
3392 IF V=6 AND S=9 AND Q=0 THEN GOSUB 708
     :GOTO 3130                                   <102>
3399 GOSUB 777                                    <169>
3400 GOTO 3130                                    <166>
3500 IF R3=0 THEN RA$(3,3)=G$(3):RA$(3,2)=
     G$(2):R3=1                                   <018>
3505 PRINT"{DOWN}AUF DEM TISCH LIEGEN VERS
     CHIEDENE SACHEN{UP}"                         <250>
3510 GOSUB 760:Q=1:RETURN                         <206>
3600 PRINT"{DOWN}ALS LETZTE EINTRAGUNG FIN
     DEN SIE:{DOWN}                               <213>
3602 PRINT"...ICH MUSS MICH VOR DEN WILDEN
      IN ACHT                                     <234>
3604 PRINT"NEHMEN.ES SIND WAHRSCHEINLICH K
     ANNIBALEN                                    <086>
3606 PRINT"FALLS SIE MICH ERWISCHEN,IST NI
     CHT NUR                                      <230>
3608 PRINT"MEINE GANZE FORSCHUNGSARBEIT UM
     SONST                                        <136>
3610 PRINT"GEWESEN,SONDERN ICH KANN AUCH N
     ICHT                                         <214>
3612 PRINT"RECHTZEITIG AM TREFFPUNKT SEIN.
     ..                                           <136>
3620 Q=1:RETURN                                   <023>
3700 IF OF=0 AND SA=1 THEN GOSUB 705:OF=1:
     RETURN                                       <075>
3703 IF OF=0 AND SA=0 THEN PRINT"{DOWN}IM
     FASS IST EIN SKORPION!ER SPRINGT":OF=
     1:GOTO 3710                                  <020>
3705 IF SA=1 AND OF=1 THEN GOSUB 709:RETUR
     N                                            <015>
3710 SA=1:PRINT"IHNEN AUF DIE HAND!!":GOSU
     B 750:D=TI:LI=2700:ZA=1:GOSUB 100            <151>
3730 IF V=9 AND S=20 THEN 3740                    <197>
3735 GOTO 900                                     <171>
3740 PRINT"{DOWN}DER SKORPION FAELLT ZU BO
     DEN UND LAEUFT                               <200>
3750 PRINT"WEG.":ZA=0:Q=1:RETURN                  <030>
3800 IF OB=1 THEN GOSUB 709:RETURN                <045>
3802 IF OB=0 THEN PRINT"{DOWN}DER BEHAELTE
     R ENTHAELT EINE ZUENDSCHNUR":OB=1            <119>
3810 IF PA=1 AND OB=1 THEN PRINT"UND SCHIE
     SSPULVER.                                    <134>
3830 Q=1:RETURN                                   <233>
3900 PRINT"{DOWN}PULVER IST IM BEHAELTER."
     :Q=1:RETURN                                  <133>
4000 PRINT SPC(9)"{GREY 1,SPACE}@@@@@@@@@@
     @@@@@@@@@@ "                                 <036>
4001 PRINT SPC(9)"H{GREEN,13SPACE}*{RVSON}
     ******{GREY 1,RVOFF}G"                       <111>
4002 PRINT SPC(9)"H{GREEN,15SPACE}*{RVSON}
     ****{GREY 1,RVOFF}G"                         <223>
4003 PRINT SPC(9)"H{GREEN,16SPACE,RVSON}**
     **{GREY 1,RVOFF}G"                           <213>
4004 PRINT SPC(9)"H{GREEN,16SPACE}*{RVSON}
     *{BROWN}*B{GREY 1,RVOFF}G"                   <055>
4005 PRINT SPC(9)"H{GREEN,18SPACE,BROWN,RV
     SON}*B{GREY 1,RVOFF}G"                       <197>
4006 PRINT SPC(9)"H{GREEN,18SPACE,BROWN,RV
     SON}TG{GREY 1,RVOFF}G"                       <182>
4007 PRINT SPC(9)"H{GREEN,18SPACE,BROWN,RV
     SON}TG{GREY 1,RVOFF}G"                       <057>
4008 PRINT SPC(9)"H{GREEN,18SPACE,BROWN,RV
     SON}HI{GREY 1,RVOFF}G"                       <129>
4009 PRINT SPC(9)"H{GREEN,17SPACE,GREY 3,S
     PACE,BROWN,RVSON}GI{GREY 1,RVOFF}G"          <061>
4010 PRINT SPC(9)"H{GREEN,2SPACE,ORANGE,SP
     ACE,GREY 3,SPACE,GREY 1}***{GREEN,10S
     PACE,GREY 3,SPACE,BROWN,RVSON}GI{GREY
```

**Listing »Odyssey« (Fortsetzung)**

```
     1,RVOFF}G"                                <045>
4011 PRINT SPC(9)"N{GREEN,SPACE,GREY 2,RVS
     ON}B{RVOFF}B{GREY 1}££££{RVSON}B{GRE
     Y 2,RVOFF}B{GREEN,3SPACE,GREY 3,4SPAC
     E,WHITE,SPACE,BROWN,RVSON,SPACE}G{GRE
     Y 1,RVOFF}G"                              <001>
4012 PRINT SPC(9)"H{GREEN,RVSON,2SPACE}U{2
     SPACE}I{5SPACE}I{4SPACE}I{3SPACE,GREY
      1,RVOFF}G"                               <075>
4013 PRINT SPC(9)"H{GREEN,RVSON,SPACE}I{4S
     PACE}U{2SPACE}U{2SPACE}K I{3SPACE}U{S
     PACE,GREY 1,RVOFF}G"                      <037>
4014 PRINT SPC(9)"H{GREEN,RVSON,2SPACE}U U
     {2SPACE}J{2SPACE}I{3SPACE}U I{3SPACE,
     GREY 1,RVOFF}G"                           <098>
4015 PRINT SPC(9)" TTTTTTTTTTTTTTTTTTTT "      <250>
4019 RA=4:OG=7:UG=7                            <231>
4020 PRINT"{GREEN}VOR IHNEN LIEGT EIN VERL
     ASSENER LAGER-                            <046>
4030 PRINT"PLATZ.IN DER FEUERSTELLE BEFIND
     ET SICH                                   <035>
4040 PRINT"ASCHE.                              <112>
4080 GOSUB 760                                 <102>
4120 PRINT"{WHITE,DOWN}MOEGLICHE RICHTUNGE
     N : N{DOWN}"                              <232>
4122 GOSUB 750                                 <130>
4125 IF AA=0 THEN GOSUB 4600                   <015>
4130 GOSUB 100                                 <010>
4135 IF S=5 THEN GOTO 4000                     <098>
4140 ON R GOTO 2500,4160,4160,4160,4160,41
     60                                        <110>
4150 GOTO 4300                                 <098>
4160 GOSUB 701                                 <120>
4300 IF V=6 AND S=7 AND R4=0 THEN RA$(4,4)
     =G$(4):R4=1:Q=1:GOSUB 760                 <205>
4499 GOSUB 777                                 <253>
4500 GOTO 4130                                 <002>
4600 PRINT"{DOWN}EIN GORILLA ERSCHEINT AUF
      DER LICHTUNG.                            <155>
4610 PRINT"ER ROLLT AUF DEM BODEN HERUM UN
     D MACHT                                   <228>
4620 PRINT"MAENNCHEN!":AA=1                    <253>
4630 GOSUB 100:IF V=12 AND S=22 THEN GOTO
     4700                                      <091>
4635 IF V=20 AND S=19 THEN GOTO 900            <249>
4637 IF R=1 THEN PRINT"{DOWN}DER GORILLA V
     ERSPERRT IHNEN DEN WEG."                  <161>
4640 PRINT"{DOWN}DER GORILLA HAT AUFGEHOER
     T UND SCHAUT                              <072>
4650 PRINT"SIE GRIMMIG AN!                     <174>
4670 GOSUB 100:IF V=12 AND S=22 THEN GOTO
     4700                                      <131>
4680 GOTO 900                                  <100>
4700 PRINT"{DOWN}DER GORILLA TIPPT SICH AN
      DEN KOPF UND                             <018>
4710 PRINT"VERSCHWINDET WIEDER IM DSCHUNGE
     L.":RETURN                                <021>
5000 PRINT SPC(9)"{GREEN,SPACE,GREY 1}@@@@
     @@@@@@@@@@@@@@@@{GREEN,SPACE}"            <190>
5001 PRINT SPC(9)"{GREY 1}H{BROWN,RVSON,SP
     ACE,RVOFF}YU{RVSON}OPOI{RVOFF}UYMF{SP
     ACE,GREEN,8SPACE,GREY 1}G"                <244>
5002 PRINT SPC(9)"H{BROWN,RVSON,SPACE,GREE
     N,RVOFF}TYUM EFRN M{BROWN,8SPACE,GREY
      1}G"                                     <225>
5003 PRINT SPC(9)"H{BROWN,RVSON,SPACE}U{RV
     OFF}TOII{RVSON}UYYU{RVOFF}P@{GREEN}TE
     DEF{3SPACE,GREY 1}G"                      <126>
5004 PRINT SPC(9)"H{BROWN,RVSON,2SPACE}@PO
     TI{RVOFF}YU{RVSON}P @POI{RVOFF}Y{GREE
     N}J{ORANGE,3SPACE,GREY 1}G"               <157>
5005 PRINT SPC(9)"H{BROWN,RVSON,SPACE,GREE
     N,RVOFF,3SPACE,YELLOW,SPACE,GREEN,5SP
     ACE}K{7SPACE,YELLOW,SPACE,GREEN,SPACE
     ,GREY 1}G"                                <125>
5006 PRINT SPC(9)"H{BROWN,RVSON}H{YELLOW,R
     VOFF}HIGG{GREEN,4SPACE,LIG.GREEN,SPAC
     E,GREEN}K{5SPACE,YELLOW}Y TY{GREY 1}G
     "                                         <220>
5007 PRINT SPC(9)"H{BROWN,RVSON,SPACE,YELL
     OW,RVOFF}BBBH{SPACE,GREEN,3SPACE,LIG.
     GREEN,SPACE,GREEN}J{4SPACE,YELLOW}HYB
     BB{GREY 1}G"                              <005>
5008 PRINT SPC(9)"H{BROWN,RVSON,SPACE,YELL
     OW,RVOFF}BHGB{SPACE,GREEN,4SPACE}G{4S
     PACE,YELLOW}HBBBB{GREY 1}G"               <234>
5009 PRINT SPC(9)"H{BROWN,RVSON}H{YELLOW,R
     VOFF}BBBBB{ORANGE,4SPACE,GREEN}G{ORAN
     GE,4SPACE,YELLOW}BBBBB{GREY 1}G"          <046>
5010 PRINT SPC(9)"H{BROWN,RVSON,SPACE,GREY
      1,RVOFF}I{RVSON}UUYYYTTT TTYYYUU{RVO
     FF}IOG"                                   <162>
5011 PRINT SPC(9)"H{BROWN,RVSON,SPACE,GREY
      1,19SPACE,RVOFF}G"                       <176>
5012 PRINT SPC(9)"H{BROWN,RVSON,SPACE,GREY
      1,RVOFF}U{RVSON}IIOOOPPPPPPOOOII{RVO
     FF}UYG"                                   <246>
5013 PRINT SPC(9)"H{BROWN,RVSON}H{GREEN,RV
     OFF,19SPACE,GREY 1}G"                     <069>
5014 PRINT SPC(9)"H{BROWN,RVSON,SPACE,GREE
     N,RVOFF,19SPACE,GREY 1}H"                 <198>
5015 PRINT SPC(9)"H{BROWN,RVSON,SPACE,GREE
     N,RVOFF,19SPACE,GREY 1}G"                 <229>
5016 PRINT SPC(9)"{GREEN,SPACE,GREY 1}TTTT
     TTTTTTTTTTTTTTTT{GREEN,SPACE}"            <174>
5019 RA=5:OG=1:UG=1                            <220>
5020 PRINT"{GREEN}SIE BEFINDEN SICH AM FUS
     S EINES BAUMES. NOERDLICH VON IHNEN L
     IEGT EIN 6                                <247>
5030 PRINT"{8LEFT}METER{3SPACE}BREITER,MIT
      SCHILF BEWACHSENER SUMPF.                <044>
5040 PRINT"AN DER MITTE EINES ASTES HAENGT
      EINE{4SPACE}LIANE MITTEN UEBER DEM S
     UMPF.                                     <034>
5050 GOSUB 760                                 <056>
5120 IF SC=0 THEN PRINT"{WHITE,DOWN}MOEGLI
     CHE RICHTUNGEN : O"                       <087>
5122 IF SC=1 THEN PRINT"{WHITE,DOWN}MOEGLI
     CHE RICHTUNGEN : N,O"                     <250>
5125 GOSUB 750                                 <117>
5130 GOSUB 100                                 <250>
5135 IF S=5 THEN GOTO 5000                     <083>
5140 ON R GOTO 5600,5160,5160,2000,5160,51
     60                                        <003>
5150 GOTO 5300                                 <090>
5160 GOSUB 701                                 <104>
5300 IF V=13 THEN 900                          <005>
5310 IF BU=0 AND S=2 AND SP$(4)<>""AND V=1
     4 THEN GOSUB 5700                         <030>
5320 IF BU=1 AND S=18 AND V=11 THEN GOSUB
     5710                                      <071>
5499 GOSUB 777                                 <237>
5500 GOTO 5130                                 <250>
5600 IF SC=1 THEN 5720                         <103>
5610 GOTO 5160                                 <200>
5700 PRINT"{DOWN}SIE HABEN PECH!DER BAUM F
     AELLT ENTLANG                             <104>
5702 PRINT"{DOWN}DES UFERS.DER AST MIT DER
      LIANE STEHT                              <132>
5704 PRINT"{DOWN}SENKRECHT IN DIE LUFT.":B
     U=1:Q=1                                   <253>
5705 PRINT"{DOWN}DAS BEIL IST ZERBROCHEN .
     ":SP$(4)="":WI=WI-1                       <060>
5706 RETURN                                    <176>
5710 PRINT"{DOWN}DIE LIANE VERFAENGT SICH
     IM SCHILF!":SC=1:Q=1:GOTO 5122            <186>
5720 PRINT"{DOWN}GESCHAFFT!SIE SIND AUF DE
     R ANDEREN SEITE                           <143>
5730 PRINT"{DOWN,UP}DIE LIANE IST ABER ABG
     ERISSEN.":GOSUB 750:GOTO 6000             <079>
6000 PRINT SPC(9)"{GREEN,SPACE,GREY 1}@@@@
     @@@@@@@@@@@@@@@@{GREEN,GREY 1}"           <028>
6002 PRINT SPC(9)"{GREY 1}H{GREEN,20SPACE,
     GREY 1}G"                                 <214>
6003 PRINT SPC(9)"N{SPACE,GREY 1,GREEN,19S
     PACE,GREY 1}G"                            <081>
6004 PRINT SPC(9)"H{GREEN,5SPACE,RVSON}£{R
     VOFF}={13SPACE,GREY 1}G"                  <249>
6005 PRINT SPC(9)"H{GREEN,3SPACE}££{RVSON,
     SPACE,RVOFF}£{SPACE,RVSON}£{RVOFF}£{1
     0SPACE,GREY 1}G"                          <079>
6006 PRINT SPC(9)"H{GREEN,2SPACE,YELLOW}Q{
     GREEN,2SPACE,RVSON}££={RVOFF,SPACE}={
     YELLOW}Q{GREEN,9SPACE,GREY 1}G"           <017>
6007 PRINT SPC(9)"H{GREEN,2SPACE}£{RVSON}£
     =£{YELLOW,RVOFF}Q{GREEN,RVSON}={RVOFF
     }££{10SPACE,GREY 1}G"                     <120>
6008 PRINT SPC(9)"H{GREEN,2SPACE}={SPACE,R
     VSON}££=£{RVOFF,2SPACE}={9SPACE,GREY
     1}G"                                      <242>
6009 PRINT SPC(9)"H{GREEN,SPACE,YELLOW}Q{S
     PACE,GREEN,RVSON}££{YELLOW,RVOFF}Q{GR
```

```
      EEN,RVSON}〒═〒{RVOFF,SPACE,YELLOW}&{GR
      EEN,9SPACE,GREY 1}〒"                  <094>
6010 PRINT SPC(9)"〒{GREEN,2SPACE,RVSON}〒{R
      VOFF,SPACE,RVSON}═〒{RVOFF,SPACE,RVSO
      N}═{RVOFF}〒{10SPACE,GREY 1}〒"         <050>
6011 PRINT SPC(9)"〒{GREEN,2SPACE,RVSON}═{R
      VOFF,SPACE,YELLOW}&{GREEN,RVSON}═══{R
      VOFF}〒 ═{9SPACE,GREY 1}〒"             <237>
6012 PRINT SPC(9)"〒{GREEN,2SPACE,YELLOW}&{
      GREEN,2SPACE,RVSON}〒═{RVOFF,2SPACE}═{
      YELLOW}&{GREEN,9SPACE,GREY 1}〒"        <100>
6013 PRINT SPC(9)"〒{GREEN,4SPACE}〒{RVSON}〒
      〒{RVOFF,2SPACE,YELLOW}&{GREEN,10SPACE
      ,GREY 1}〒"                             <144>
6014 PRINT SPC(9)"〒{BROWN,RVSON}YTU{3SPACE
      }ITYUYTU ITYUUY{GREY 1,RVOFF}〒"        <218>
6015 PRINT SPC(9)"〒{BROWN,RVSON,SPACE}J{2S
      PACE}U{5SPACE}I{4SPACE}K{4SPACE,GREY
      1,RVOFF}〒"                             <025>
6016 PRINT SPC(9)"〒{BROWN,RVSON,3SPACE}K{3
      SPACE}U{5SPACE}I{3SPACE}J{2SPACE,GREY
       1,RVOFF}〒"                            <254>
6017 PRINT SPC(9)"{BROWN,SPACE,GREY 1}TTTT
      TTTTTTTTTTTTTTTT{BROWN,SPACE}"          <073>
6019 RA=6:OG=12:UG=11                        <053>
6020 PRINT"{GREEN}SIE STEHEN VOR EINEM WIL
      DSTRAUCH.                               <166>
6030 GOSUB 760                               <020>
6120 PRINT"{WHITE,DOWN}MOEGLICHE RICHTUNGE
      N : W,O"                                <217>
6130 GOSUB 100                               <232>
6135 IF S=5 THEN GOTO 6000                   <066>
6140 ON R GOTO 6160,6160,7000,8000,6160,61
      60                                      <109>
6150 GOTO 6300                               <082>
6160 GOSUB 701                               <088>
6300 IF V=6 AND S=12 THEN RA$(6,5)=G$(5):G
      OSUB 760:Q=1                            <070>
6499 GOSUB 777                               <221>
6500 GOTO 6130                               <242>
7000 PRINT SPC(9)"{GREY 1,SPACE}〒〒〒〒〒〒〒〒〒〒
      〒〒〒〒〒〒〒〒〒〒 "                        <244>
7001 PRINT SPC(9)"〒{BLUE,14SPACE,GREEN,6SP
      ACE,GREY 1}〒"                          <069>
7002 PRINT SPC(9)"〒{BLUE,2SPACE,GREEN,2SPA
      CE,BLUE,12SPACE,ORANGE}〒PRF{GREY 1}〒"  <225>
7003 PRINT SPC(9)"〒{BLUE,SPACE,GREEN,SPACE
      ,BLUE,6SPACE,ORANGE}RRF*CDET{GREEN,2S
      PACE,GREY 1}.{GREEN,SPACE,GREY 1}〒"    <046>
7004 PRINT SPC(9)"〒{ORANGE}RRF*CDET{GREY 1
      }.{ORANGE,SPACE,BLUE,SPACE,GREY 1}.{B
      LUE,SPACE,GREY 1}.{BLUE,SPACE,GREY 1}
      .{BLUE,SPACE,GREY 1}.{BLUE,SPACE,GREE
      N,SPACE,GREY 1}〒"                      <169>
7005 PRINT SPC(9)"〒.{GREEN,2SPACE,GREY 1}.
      {GREY 3,SPACE,GREY 1}.{ORANGE,SPACE,G
      REY 3,SPACE,ORANGE,SPACE,GREY 3,SPACE
      ,GREY 1}.{GREY 3,SPACE,GREY 1}.{GREY
      3,5SPACE,GREY 1}.{GREY 3,SPACE,GREY 1
      }〒"                                    <156>
7006 PRINT SPC(9)"〒{GREEN,SPACE,GREY 1}.{G
      REEN,2SPACE,GREY 3,SPACE,GREEN,SPACE,
      GREY 1}.{GREY 3,SPACE,GREY 1}.{GREY 3
      ,4SPACE,GREEN,SPACE,GREY 1}.{GREEN,2S
      PACE,GREY 1}.{GREEN,2SPACE,GREY 1}〒"   <063>
7007 PRINT SPC(9)"〒{GREEN,3SPACE,GREY 3,SP
      ACE,GREY 1}.{GREY 3,RVSON}£{4SPACE,RV
      OFF}£{GREEN,SPACE,GREY 1}.{SPACE,GREE
      N,SPACE,GREY 1}.{GREEN,4SPACE,GREY 1}
      〒"                                     <160>
7008 PRINT SPC(9)"〒{GREEN,SPACE,GREY 1,SPA
      CE,GREY 3,SPACE,GREY 1}.{GREEN,SPACE,
      GREY 3}OYYYY{RVSON,SPACE,RVOFF,SPACE,
      GREEN,SPACE,GREY 1}.{GREEN,4SPACE,GRE
      Y 1,SPACE}.〒"                          <138>
7009 PRINT SPC(9)"〒{GREEN,SPACE,GREY 1}.{G
      REEN,2SPACE,ORANGE,SPACE,GREY 3}〒{SPA
      CE,RED}±{GREY 3,2SPACE,RVSON,SPACE,GR
      EEN,RVOFF,5SPACE,GREY 1}.{GREEN,SPACE
      ,GREY 1}.{GREEN,SPACE,GREY 1}〒"        <196>
7010 PRINT SPC(9)"〒{GREY 3,SPACE,GREY 1,SP
      ACE,GREEN,3SPACE,GREY 3}LPPPP£{GREEN,
      SPACE,GREY 1}.{GREEN,SPACE,GREY 1}.{G
      REEN,2SPACE,ORANGE}〒PO{GREY 1}〒"      <245>
7011 PRINT SPC(9)"〒{ORANGE,SPACE,GREY 1}.{
      ORANGE,SPACE,GREY 1}.{ORANGE,4SPACE,G
      REY 1}.{ORANGE}〒POT{RVSON}UTT .{2SPAC
      E,GREY 1,RVOFF}〒"                      <246>
7012 PRINT SPC(9)"〒{ORANGE,SPACE}〒POT{RVSO
      N}UYY{3SPACE}.{2SPACE}.{3SPACE}.{SPAC
      E,GREY 1,RVOFF}〒"                      <147>
7013 PRINT SPC(9)"〒{ORANGE,RVSON,2SPACE}.{
      3SPACE}.{2SPACE}.{2SPACE}.{3SPACE}.{3
      SPACE,GREY 1,RVOFF}〒"                  <121>
7014 PRINT SPC(9)"〒{ORANGE,RVSON,4SPACE}.{
      4SPACE}.{4SPACE}.{3SPACE}.{SPACE,GREY
       1,RVOFF}〒"                            <214>
7015 PRINT SPC(9)" TTTTTTTTTTTTTTTTTTTT{OR
      ANGE,SPACE}"                            <231>
7019 RA=7:OG=1:UG=1                          <252>
7020 PRINT"{GREEN}SIE SIND AN DER WESTKUES
      TE DER INSEL.                           <068>
7030 PRINT"AM STRAND STEHT EINE AN LAND GE
      SCHWEMMTE                               <050>
7040 PRINT"{UP}METALLKISTE.                  <177>
7050 GOSUB 760                               <024>
7120 PRINT"{WHITE,DOWN}MOEGLICHE RICHTUNGE
      N : O"                                  <033>
7125 GOSUB 750                               <083>
7130 GOSUB 100                               <216>
7135 IF S=5 THEN GOTO 7000                   <051>
7140 ON R GOTO 7160,7160,7160,6000,7160,71
      60                                      <176>
7150 GOTO 7300                               <072>
7160 GOSUB 701                               <070>
7300 IF V=7 AND S=15 AND OK=1 THEN GOSUB 7
      09                                      <249>
7310 IF V=7 AND S=15 AND OK=0 THEN GOSUB 7
      600:OK=1:Q=1                            <090>
7320 IF V=10 AND S=15 AND OK=0 THEN GOSUB
      710                                     <158>
7330 IF V=10 AND S=15 AND OK=1 THEN GOSUB
      705:OK=0                                <222>
7340 IF V=15 AND S=15 AND VS=0 THEN GOTO 7
      700                                     <050>
7350 IF S=23 AND V=20 AND OK=1 AND VS=0 TH
      EN GOTO 900                             <150>
7360 IF S=23 AND VS=0 AND V=11 AND OK=1 AN
      D KG=1 THEN GOSUB 7800:Q=1              <077>
7499 GOSUB 777                               <205>
7500 GOTO 7130                               <234>
7600 IF VS=0 THEN PRINT"{WHITE,DOWN}AUF DE
      M INHALT DER KISTE SITZT EINE "         <154>
7610 IF VS=0 THEN PRINT"GROSSE,SCHWARZE VO
      GELSPINNE!":RETURN                      <209>
7620 GOSUB 705:OK=1:RETURN                   <248>
7700 PRINT"{DOWN}DER INHALT UND DIE SPINNE
       FALLEN HERAUS.                         <182>
7702 PRINT"{UP}DIE VOGELSPINNE KRABBELT AN
       IHREM BEIN                             <197>
7710 PRINT"HOCH!!":D=TI:LI=500:ZA=1:GOSUB
      100:GOTO 900                            <210>
7800 PRINT"{DOWN}DIE SPINNE LAEUFT DAVON!"
      :VS=1                                   <207>
7810 RA$(7,6)=G$(6):RA$(7,7)=G$(7):GOSUB 7
      60:RETURN                               <164>
8000 PRINT SPC(9)"{ORANGE,SPACE,GREY 1}〒〒〒
      〒〒〒〒〒〒〒〒〒〒〒〒〒〒{ORANGE,SPACE}"     <172>
8001 PRINT SPC(9)"{GREY 1}〒{RVSON,2SPACE,R
      VOFF}〒{RVSON,2SPACE}〒{5SPACE}〒{3SPACE
      }〒{2SPACE}〒{SPACE,RVOFF}〒"            <081>
8002 PRINT SPC(9)"〒{RVSON,2SPACE,RVOFF}T{R
      VSON,2SPACE}Y{SPACE,GREY 2}JK{GREY 1,
      2SPACE}S{2SPACE}〒 〒 S{SPACE,RVOFF}〒"  <191>
8003 PRINT SPC(9)"〒{RVSON,2SPACE}S{2SPACE}
      HB{4SPACE}T{2SPACE}IT〒 B{SPACE,RVOFF}
      〒"                                     <027>
8004 PRINT SPC(9)"〒{RVSON,2SPACE}B{2SPACE}
      ═ 〒{SPACE,GREY 2}JK{GREY 1,SPACE}〒 SH
      〒 〒{SPACE,RVOFF}〒"                    <073>
8005 PRINT SPC(9)"〒{RVSON,2SPACE}I〒 B B{6S
      PACE}B 〒 Y{SPACE,RVOFF}〒"              <154>
8006 PRINT SPC(9)"〒{RVSON,SPACE}HGB T H{2S
      PACE,GREY 2}JK{GREY 1,2SPACE,RVOFF}〒{
      RVSON,SPACE}S H{SPACE,RVOFF}〒"         <243>
8007 PRINT SPC(9)"〒{RVSON,SPACE}NT BH{8SPA
      CE}HH{2SPACE}H{SPACE,RVOFF}〒"           <187>
8008 PRINT SPC(9)"〒{RVSON,SPACE}IH YT{SPAC
      E,GREY 2}JK{GREY 1,6SPACE}〒{2SPACE}&{
      SPACE,RVOFF}〒"                         <131>
8009 PRINT SPC(9)"〒{RVSON,SPACE,RVOFF}〒{RV
      SON}〒 I〒{9SPACE}〒{2SPACE}I{SPACE,RVOF
```

**Listing »Odyssey« (Fortsetzung)**

```
     F}G"                                       <142>
8010 PRINT SPC(9)"R{RVSON,SPACE,RVOFF}T{RV
     SON,2SPACE}G{5SPACE,GREY 2}JK{GREY 1,
     4SPACE}G I{SPACE,RVOFF}G"                  <096>
8011 PRINT SPC(9)"R{RVSON,SPACE}B{2SPACE}G
     {10SPACE}HG R{SPACE,RVOFF}G"               <084>
8012 PRINT SPC(9)"R{RVSON,SPACE}G NG{3SPAC
     E,GREY 2}JK{GREY 1,5SPACE}-B NG{RVOFF
     }G"                                        <021>
8013 PRINT SPC(9)"R{RVSON,SPACE}G NB{3SPAC
     E}R{2SPACE,GREY 2}JK{GREY 1,2SPACE}G{
     RVOFF}R{RVSON}R G{RVOFF}G"                 <210>
8014 PRINT SPC(9)"R{RVSON,SPACE}I Y G{2SPA
     CE,RVOFF}R{RVSON,6SPACE}GMM G{RVOFF}G
     "                                          <148>
8015 PRINT SPC(9)"R{RVSON,SPACE}G H{SPACE,
     RVOFF}T{RVSON,2SPACE,RVOFF}T{RVSON,5S
     PACE}HG{SPACE,RVOFF}N{RVSON,SPACE}B{R
     VOFF}G"                                    <195>
8016 PRINT SPC(9)" TTTTTTTTTTTTTTTTTTTT ".     <187>
8019 RA=8:OG=1:UG=1                             <013>
8020 PRINT"{GREEN}SIE STEHEN VOR EINER FEL
     SWAND MIT                                  <071>
8030 PRINT"SPALTEN ,AN DER FUSSTRITTE SIND
                                                <196>
8050 GOSUB 760                                  <008>
8120 PRINT"{WHITE,DOWN}MOEGLICHE RICHTUNGE
     N : W,HINAUF"                              <202>
8130 GOSUB 100                                  <200>
8135 IF S=5 THEN GOTO 8000                      <036>
8140 ON R GOTO 8160,8160,6000,8160,8600,81
     60                                         <032>
8150 GOTO 8299                                  <147>
8160 GOSUB 701                                  <054>
8299 GOSUB 777                                  <142>
8300 GOTO 8130                                  <033>
8600 IF FH=1 THEN GOTO 9000                     <114>
8602 IF FH=0 THEN PRINT"{DOWN,WHITE}SIE ST
     EIGEN DIE WAND HINAUF.                     <074>
8610 IF FH=0 THEN PRINT"{DOWN}PLOETZLICH K
     OMMT VON OBEN EIN HAGEL VON                <091>
8620 IF FH=0 THEN PRINT"FELSBROCKEN AUF SI
     E ZU!!":FH=1:D=TI:LI=3600:ZA=1             <157>
8625 GOSUB 100                                  <187>
8630 IF S=33 THEN PRINT"{DOWN}DER STEINHAG
     EL FAELLT AN IHNEN VORBEI!":GOTO 8650      <191>
8640 GOTO 900                                   <250>
8650 ZA=0:GOSUB 100                             <085>
8660 IF V=18 THEN GOSUB 705:GOTO 8700           <153>
8665 IF V=13 THEN GOTO 8680                     <128>
8670 GOTO 8650                                  <028>
8680 RA=9:GOTO 900                              <122>
8700 PRINT"{DOWN}SIE STEHEN WIEDER IN DER
     WAND."                                     <152>
8710 GOSUB 100                                  <018>
8720 ON R GOTO 8730,8730,8730,8730,9000,80
     00                                         <218>
8725 GOTO 8735                                  <102>
8730 GOSUB 701                                  <118>
8735 GOSUB 777                                  <171>
8740 GOTO 8710                                  <244>
9000 PRINT SPC(9)"{GREEN,SPACE,GREY 1}GGGG
     GGGGGGGGGGGGGGGG{GREEN,SPACE}"             <126>
9001 PRINT SPC(9)"{GREY 1}R{GREY 3,5SPACE,
     GREY 2,RVSON}£N//NN / /N{2SPACE}N{SPA
     CE,GREY 1,RVOFF}G"                         <054>
9002 PRINT SPC(9)"R{GREY 3,4SPACE,GREY 2,R
     VSON}£NFDDYDDCFN{2SPACE}N{2SPACE,GREY
      1,RVOFF}G"                                <253>
9003 PRINT SPC(9)"R{GREY 3,4SPACE,GREY 2,R
     VSON}EY{2SPACE}PTOP{2SPACE}B N{3SPACE
     ,GREY 1,RVOFF}G"                           <231>
9004 PRINT SPC(9)"R{GREY 3,3SPACE,GREY 2,R
     VSON}£{2SPACE,RVOFF}Q{SPACE,BLUE,SPAC
     E}Y{SPACE,WHITE}={GREY 2}T{RVSON,2SPA
     CE}G{3SPACE}N{GREY 1,RVOFF}G"              <087>
9005 PRINT SPC(9)"R{GREY 3,2SPACE,GREY 2}B
     {RVSON,2SPACE,RVOFF}V{2SPACE,BLUE,SPA
     CE}I{GREY 2,SPACE,WHITE}ZL{GREY 2}T{R
     VSON,SPACE}H{2SPACE}N{SPACE,GREY 1,RV
     OFF}G"                                     <251>
9006 PRINT SPC(9)"R{GREY 3,2SPACE,GREY 2,R
     VSON}R T{RVOFF,3SPACE,BLUE}RGGGGG{GRE
     Y 2,RVSON,2SPACE}GN{2SPACE,GREY 1,RVO
     FF}G"                                      <031>
9007 PRINT SPC(9)"R{GREY 3,2SPACE,GREY 2,R
     VSON,2SPACE,RVOFF,4SPACE,BLUE}/{SPACE
     ,YELLOW,RVSON}£X{RVOFF}£M{GREY 2,RVSO
     N}T B{2SPACE}/{GREY 1,RVOFF}G"             <063>
9008 PRINT SPC(9)"R{GREY 3,SPACE,GREY 2}R{
     RVSON,SPACE}T{YELLOW,RVOFF,3SPACE,BLU
     E}/{SPACE,YELLOW,RVSON}£TTF{RVOFF}N{G
     REY 2}R{RVSON,SPACE}H /{SPACE,GREY 1,
     RVOFF}G"                                   <063>
9009 PRINT SPC(9)"R{LIG.RED,SPACE,GREY 2}T
     {RVSON,SPACE,RVOFF}R{YELLOW,2SPACE,BL
     UE}/{8SPACE,GREY 2,RVSON,SPACE}R/{2SP
     ACE,GREY 1,RVOFF}G"                        <074>
9010 PRINT SPC(9)"R{LIG.RED,SPACE,GREY 2}T
     {RVSON,SPACE,RVOFF}T{YELLOW,SPACE,BLU
     E}/{GREY 1,9SPACE,GREY 2,RVSON}R G{2S
     PACE,GREY 1,RVOFF}G"                       <210>
9011 PRINT SPC(9)"R{LIG.RED,SPACE,GREY 2,R
     VSON}R{SPACE,RVOFF}G{BLUE}/{GREY 1,SP
     ACE,BLUE,6SPACE,GREY 2,3SPACE,RVSON}T
      R /{GREY 1,RVOFF}G"                       <049>
9012 PRINT SPC(9)"R{LIG.RED,SPACE,GREY 2,R
     VSON}R{SPACE,BLUE,RVOFF}/{GREY 1,8SPA
     CE,BLUE,2SPACE,GREY 2,SPACE,RVSON}R G
     /{SPACE,GREY 1,RVOFF}G"                    <058>
9013 PRINT SPC(9)"R{RVSON}£,{2SPACE}. ,{3S
     PACE},{2SPACE}. , . ,{RVOFF}G"             <182>
9014 PRINT SPC(9)"R{RVSON}.{2SPACE}, .{2SP
     ACE}, . ,{3SPACE}.{2SPACE,RVOFF}£G"        <238>
9015 PRINT SPC(9)"R{RVSON,3SPACE}.{2SPACE}
     , .{2SPACE}.{4SPACE},{SPACE.RVOFF}£ G
     "                                          <130>
9016 PRINT SPC(9)" TTTTTTTTTTTTTTTTTTTT "      <171>
9019 RA=9:OG=1:UG=1                             <029>
9020 PRINT"{GREEN}SIE STEHEN VOR DEM EINGA
     NG EINER HOEHLE                            <147>
9050 GOSUB 760                                  <248>
9120 PRINT"{WHITE,DOWN}MOEGLICHE RICHTUNGE
     N : N,O,RUNTER"                            <195>
9130 GOSUB 100                                  <184>
9135 IF S=5 THEN GOTO 9000                      <021>
9140 ON R GOTO 10000,9160,9160,11000,9160,
     8000                                       <213>
9150 GOTO 9299                                  <139>
9160 GOSUB 701                                  <038>
9299 GOSUB 777                                  <227>
9300 GOTO 9130                                  <016>
10000 PRINT SPC(9)"{GREY 1,SPACE}GGGGGGGGG
      GGGGGGGGGGG"                              <196>
10001 PRINT SPC(9)"R{BLUE,7SPACE,WHITE,SPA
      CE}G{BLUE,8SPACE}N{2SPACE,GREY 1}G"       <218>
10002 PRINT SPC(9)"R{WHITE,RVSON}EE{RVOFF,
      3SPACE,RVSON}EE{RVOFF,SPACE}B{8SPACE
      ,BLUE}R{WHITE,2SPACE,GREY 1}G"            <147>
10003 PRINT SPC(9)"R{BLUE,2SPACE,WHITE,RVS
      ON}£ R{BLUE,RVOFF,3SPACE,WHITE}G{BLU
      E,4SPACE,WHITE,SPACE,BLUE,3SPACE}N{2
      SPACE,GREY 1}G"                           <216>
10004 PRINT SPC(9)"R{WHITE,SPACE}M{3SPACE}
      N B{9SPACE,BLUE}R{WHITE,2SPACE,GREY
      1}G"                                      <219>
10005 PRINT SPC(9)"R{WHITE}G{SPACE,RVSON}R
      RR{BLUE,RVOFF,2SPACE,WHITE}G{BLUE,3S
      PACE,WHITE,SPACE,BLUE,5SPACE}N{2SPAC
      E,GREY 1}G"                               <226>
10006 PRINT SPC(9)"R{WHITE}-{GREY 1}R{WHIT
      E,RVSON}EEE{RVOFF,SPACE}B{BLUE,2SPAC
      E,WHITE,2SPACE,BLUE,6SPACE}N{SPACE,W
      HITE,SPACE,GREY 1}G"                      <229>
10007 PRINT SPC(9)"R{WHITE}RN{3SPACE}MG{10
      SPACE,BLUE}R{WHITE,2SPACE,GREY 1}G"       <034>
10008 PRINT SPC(9)"R{BLUE,SPACE,WHITE}EF{B
      LUE,SPACE,WHITE}FE{YELLOW,SPACE}TDET
      TEDT{BLUE,SPACE,YELLOW,SPACE,BLUE}N{
      YELLOW,2SPACE,GREY 1}G"                   <251>
10009 PRINT SPC(9)"R{BLUE,3SPACE,WHITE}T{B
      LUE,2SPACE,YELLOW,SPACE}MEDCCDEN{BLU
      E,SPACE,YELLOW,SPACE,BLUE}N{2SPACE,G
      REY 1}G"                                  <073>
10010 PRINT SPC(9)"R{BLUE,6SPACE,YELLOW,2S
      PACE}MPGGPN{SPACE,BLUE,SPACE,YELLOW,
      SPACE,BLUE}N{2SPACE,GREY 1}G"             <056>
10011 PRINT SPC(9)"R{BLUE}PPPPPPP{YELLOW,R
      VSON}£{SPACE,RVOFF}L{2SPACE}G{RVSON,
      SPACE,RVOFF}£M{BLUE}PB{2SPACE,GREY 1
      }G"                                       <193>
10012 PRINT SPC(9)"R{BLUE,5SPACE,YELLOW,SP
```

```
      ACE,RVSON}£{6SPACE,RVOFF}£{2SPACE}M{
      BLUE,SPACE}M{SPACE,GREY 1}G"              <012>
10013 PRINT SPC(9)"R{3SPACE,BLUE,SPACE,YEL
      LOW,SPACE,RVSON}£TTTTTTTT*{RVOFF,3SPA
      CE}M{BLUE,SPACE}M{GREY 1}G"               <224>
10014 PRINT SPC(9)"R{BLUE,3SPACE,YELLOW,SP
      ACE,RVSON}£ ZZM MZZ *{BLUE,RVOFF,SPA
      CE,YELLOW,SPACE}N{BLUE,2SPACE,GREY 1
      }G"                                       <161>
10015 PRINT SPC(9)"R{BLUE,2SPACE,YELLOW,SP
      ACE,RVSON}£{5SPACE}M{5SPACE}*{RVOFF}
      N{BLUE,3SPACE,GREY 1}G"                   <132>
10016 PRINT SPC(9)"R{20SPACE}G"                 <112>
10017 PRINT SPC(9)"{BLUE,SPACE,GREY 1}TTTT
      TTTTTTTTTTTTTTTTTT{BLUE,SPACE}"           <255>
10019 RA=10:OG=13:UG=13                         <192>
10020 PRINT"{GREEN}SIE STEHEN IN EINEM GOE
      TZENTEMPEL.                               <007>
10030 PRINT"AUF DEM ALTAR BEFINDET SICH EI
      NE SCHALE                                 <221>
10040 PRINT"AUS GOLD.                           <155>
10100 GOSUB 760                                 <026>
10120 PRINT"{WHITE,DOWN}MOEGLICHE RICHTUNG
      EN : S{DOWN}"                             <154>
10125 GOSUB 750                                 <035>
10130 GOSUB 100                                 <168>
10135 IF S=5 THEN GOTO 10000                    <000>
10140 ON R GOTO 10160,9000,10160,10160,101
      60,1160                                   <204>
10150 GOTO 10300                                <048>
10160 GOSUB 701                                 <022>
10300 IF V=6 AND S=13 THEN GOSUB 10600          <195>
10310 IF V=1 AND SP$(8)<>""AND WA=0 THEN G
      OSUB 10700                                <146>
10499 GOSUB 777                                 <157>
10500 GOTO 10130                                <018>
10600 IF R8=0 THEN PRINT"{DOWN}DIE SCHALE
      ENTHAELT EDELSTEINE.                      <003>
10610 IF R8=0 THEN RA$(10,8)=G$(8):GOSUB 7
      60:R8=1                                   <182>
10620 Q=1:RETURN                                <165>
10700 IF WA=0 THEN GOSUB 10705                  <096>
10702 Q=1:RETURN                                <247>
10705 PRINT"{DOWN}DER ALTER BEGINNT IM BOD
      EN ZU VERSCHWIN-                          <100>
10710 PRINT"{UP}DEN.VOR DEN AUSGANG SCHIEB
      T SICH EINE                               <067>
10720 PRINT"MAUER!DIE WAENDE BEGINNEN SICH
       AUF EIN-                                 <143>
10730 PRINT"ANDER ZU ZUBEWEGEN!!":WA=1:D=T
      I:LI=2400:ZA=1                            <161>
10740 PRINT"{DOWN}MOEGLICHE RICHTUNGEN: KE
      INE                                       <059>
10750 PRINT"{DOWN}SIE SIND GEFANGEN!            <056>
10760 GOSUB 100                                 <036>
10770 ON R GOTO 10790,10790,10790,10790,10
      790,10790                                 <186>
10780 GOTO 10800                                <077>
10790 GOSUB 701                                 <146>
10800 IF V=23 AND S=25 THEN GOTO 10900          <009>
10805 IF T=1 THEN GOTO 900                      <176>
10810 GOSUB 777                                 <214>
10820 GOTO 10760                                <213>
10900 ZA=0:T=0                                  <086>
10910 PRINT"{DOWN}DIE WAENDE GEHEN ZURUECK
      .DER AUSGANG ISTFREI!":GOTO 10120         <064>
11000 PRINT SPC(9)"{GREEN,SPACE,GREY 1}@@@
      @@@@@@@@@@@@@@@@@ "                       <085>
11001 PRINT SPC(9)"R{GREEN}PPPPPPPPPPPP{8S
      PACE,GREY 1}G"                            <019>
11002 PRINT SPC(9)"R{GREEN,11SPACE}NM{7SPA
      CE,GREY 1}G"                              <072>
11003 PRINT SPC(9)"R{ORANGE}I{GREEN,SPACE,
      ORANGE}U{2SPACE}U{GREEN,2SPACE,ORANG
      E}J{GREEN,SPACE}N{SPACE,GREY 2,SPACE
      ,GREEN}M{6SPACE,GREY 1}G"                 <107>
11004 PRINT SPC(9)"R{GREEN,SPACE,ORANGE}U{
      2SPACE,GREEN,2SPACE,ORANGE}U{GREEN,2
      SPACE}N{2SPACE,GREY 2,SPACE,GREEN,SP
      ACE}M{5SPACE,GREY 1}G"                    <139>
11005 PRINT SPC(9)"R{ORANGE}XXXXXXXX{GREY
      1}PP{GREEN,5SPACE}M{4SPACE,GREY 1}G"      <089>
11006 PRINT SPC(9)"R{GREEN,8SPACE,GREY 1,S
      PACE}NR{GREEN,5SPACE}M{3SPACE,GREY 1
      }G"                                       <160>
11007 PRINT SPC(9)"R{GREEN,8SPACE,GREY 1}N
      {GREEN,SPACE,GREY 1}LP{GREEN,5SPACE}
      M{2SPACE,GREY 1}G"                        <040>
11008 PRINT SPC(9)"R{GREEN,5SPACE,GREY 1,2
      SPACE}N{GREEN,SPACE,GREY 1}/{GREEN,S
      PACE,GREY 1}NR{GREEN,5SPACE}M{SPACE,
      GREY 1}G"                                 <254>
11009 PRINT SPC(9)"W{ORANGE}XXXX{GREEN,SPA
      CE,GREY 1,SPACE}N{GREEN,SPACE,GREY 1
      }/{GREEN,SPACE,GREY 1}N{GREEN,SPACE,
      GREY 1}LP{GREEN,5SPACE}M{GREY 1}G"        <225>
11010 PRINT SPC(9)"W{ORANGE}I{GREEN,2SPACE
      }N{GREY 1}YP /{GREEN,SPACE,GREY 1}N{
      GREEN,SPACE,GREY 1}/ NR{GREEN,5SPACE
      ,GREY 1}G"                                <155>
11011 PRINT SPC(9)"W{GREEN,SPACE,ORANGE}U{
      GREEN}N{WHITE,SPACE,GREY 2,SPACE,GRE
      Y 1}W/ N /{GREEN,SPACE,GREY 1}N LP{G
      REEN,4SPACE,GREY 1}G"                     <174>
11012 PRINT SPC(9)"W{GREEN,SPACE}N{LIG.BLU
      E,SPACE,GREEN,SPACE,GREY 2,SPACE,GRE
      EN,SPACE,GREY 1}YP /{GREEN,SPACE,GRE
      Y 1}N / NR{GREEN,3SPACE,GREY 1}G"         <223>
11013 PRINT SPC(9)"W{GREEN}N{GREY 1,SPACE,
      LIG.BLUE,SPACE,GREEN,SPACE,GREY 2,SP
      ACE,GREEN,2SPACE,GREY 1}R/ N / N{GRE
      EN,SPACE,GREY 1}LP{GREEN,2SPACE,GREY
       1}G"                                     <009>
11014 PRINT SPC(9)"W{GREEN,4SPACE,GREY 2,S
      PACE,GREEN,3SPACE,GREY 1}YP / N / NR
      {GREEN,SPACE,GREY 1}G"                    <025>
11015 PRINT SPC(9)"R{GREEN,4SPACE,GREY 2,S
      PACE,GREEN,4SPACE,GREY 1}W/ N{GREEN,
      SPACE,GREY 1}/{GREEN,SPACE,GREY 1}N{
      GREEN,SPACE,GREY 1}L{GREEN,SPACE,GRE
      Y 1}G"                                    <246>
11016 PRINT SPC(9)"{GREEN,SPACE,GREY 1}TTT
      TTTTTTTTTTTTTTTTTTT{GREEN,SPACE}"         <078>
11019 RA=11:OG=1:UG=1                           <070>
11020 PRINT"{GREEN}VOR IHNEN FUEHRT EINE T
      REPPE AUF EINER                           <144>
11021 PRINT"SCHRAEGEN RAMPE NACH UNTEN.         <070>
11050 GOSUB 760                                 <216>
11120 PRINT"{WHITE,DOWN}MOEGLICHE RICHTUNG
      EN : W,RUNTER"                            <078>
11125 GOSUB 750                                 <019>
11130 GOSUB 100                                 <152>
11135 IF S=5 THEN GOTO 11000                    <242>
11136 IF V=24 AND S=26 AND R=6 THEN GOTO 1
      1500                                      <179>
11137 IF V=11 AND G=8 AND SP$(8)<>""AND R=
      6 THEN GOSUB 705:GOSUB 11450              <167>
11138 IF V=11 AND R=6 OR G<>0 AND R=6 THEN
       R=0                                      <004>
11140 ON R GOTO 11160,11160,9000,11160,111
      60,11400                                  <212>
11150 GOTO 11299                                <214>
11160 GOSUB 701                                 <006>
11299 GOSUB 777                                 <195>
11300 GOTO 11130                                <072>
11400 PRINT"{DOWN}SIE GEHEN DIE TREPPE HIN
      UNTER.                                    <048>
11410 PRINT"{DOWN}PLOETZLICH BEGINNT DIE T
      REPPE UNTER                               <177>
11420 PRINT"{DOWN}IHNEN EINZUSTUERZEN!!         <129>
11430 D=TI:LI=600:ZA=1:GOSUB 100                <218>
11435 IF V=13 THEN ZA=0:T=0:GOTO 11500          <234>
11440 GOTO 900                                  <000>
11450 SP$(8)="":RA$(18,8)=G$(8):WI=WI-1:R=
      0:RETURN                                  <201>
11500 REM *** RAUM 11500 ****************
      ***********************************
      **                                        <240>
11519 RA=18:OG=1:UG=1                           <254>
11520 PRINT"{GREEN,2DOWN}SIE BEFINDEN SICH
       AM FUSS DER TREPPE.                      <237>
11530 PRINT"EIN PFAD VERLAEUFT NACH NORDEN
                                                <152>
11550 GOSUB 760                                 <208>
11600 PRINT"{WHITE,DOWN}MOEGLICHE RICHTUNG
      EN : N{DOWN}"                             <090>
11610 GOSUB 100                                 <124>
11620 ON R GOTO 12000,11640,11640,11640,11
      640,11640                                 <182>
11625 IF S=5 THEN GOTO 11500                    <244>
11630 GOTO 11645                                <247>
11640 GOSUB 701                                 <234>
```

**Listing »Odyssey« (Fortsetzung)**

```
11645 GOSUB 777                                    <031>
11650 GOTO 11610                                   <200>
12000 PRINT SPC(9)"{ORANGE,SPACE,GREY 1}GG
      GGGGGGGGGGGGGGGGGG{ORANGE,SPACE}"           <106>
12001 PRINT SPC(9)"{GREY 1}R{ORANGE,SPACE,
      GREEN}B{RVSON}TTTTTTB{RVOFF}B{ORANGE
      ,10SPACE,GREY 1}G"                          <177>
12002 PRINT SPC(9)"R{GREEN,RVSON}BB{RVOFF}
      T{RVSON}TTTT{RVOFF}T{RVSON}B{RVOFF}B
      B{ORANGE,9SPACE,GREY 1}G"                   <001>
12003 PRINT SPC(9)"R{GREEN,SPACE}B{RVSON}T
      TTTTTBB{RVOFF,10SPACE,GREY 1}G"             <173>
12004 PRINT SPC(9)"R{GREEN,RVSON}BT{RVOFF}
      ={RVSON}TTTT{RVOFF}={RVSON}T{RVOFF}B
      B{9SPACE,GREY 1}G"                          <011>
12005 PRINT SPC(9)"R{GREEN}B{RVSON}BBB{BRO
      WN}T{GREEN,RVOFF}B{BROWN}T{GREEN}BBB
      {10SPACE,GREY 1}G"                          <037>
12006 PRINT SPC(9)"R{GREEN,SPACE}B{RVSON}B
      {RVOFF,SPACE}B{BROWN,RVSON}B{GREEN}B
      {RVOFF,SPACE}B{RVSON}B{RVOFF}B{9SPAC
      E,GREY 1}G"                                 <250>
12007 PRINT SPC(9)"R{GREEN,4SPACE,BROWN,RV
      SON}TBT{GREEN,RVOFF,13SPACE,GREY 1}G
      "                                           <107>
12008 PRINT SPC(9)"R{GREEN,4SPACE,BROWN,RV
      SON}TBT{GREEN,RVOFF,9SPACE,ORANGE,4S
      PACE,GREY 1}G"                              <089>
12009 PRINT SPC(9)"R{GREEN,4SPACE,BROWN,RV
      SON}TBT{GREEN,RVOFF,13SPACE,GREY 1}G
      "                                           <109>
12010 PRINT SPC(9)"R{ORANGE}G{GREEN,3SPACE
      ,BROWN,RVSON}TST{GREEN,RVOFF,13SPACE
      ,GREY 1}G"                                  <245>
12011 PRINT SPC(9)"R{ORANGE}TYEB{BROWN,RVS
      ON}HSH{ORANGE,RVOFF}CDEYEDC*FRFCD{GR
      EY 1}G"                                     <202>
12012 PRINT SPC(9)"R{GREEN,4SPACE,BROWN,RV
      SON}HIN{GREEN,RVOFF,13SPACE,GREY 1}G
      "                                           <008>
12013 PRINT SPC(9)"R{ORANGE,RVSON,SPACE}TY
      T.{2SPACE}.TYU{RVOFF}TOT{RVSON}U{RVO
      FF}TOT{RVSON}UY{GREY 1,RVOFF}R"             <221>
12014 PRINT SPC(9)"R{ORANGE,RVSON,SPACE}.{
      3SPACE}.{4SPACE}.{3SPACE}.{4SPACE}.{
      GREY 1,RVOFF}G"                             <040>
12015 PRINT SPC(9)"R{ORANGE,RVSON,4SPACE}.
      {3SPACE}.{4SPACE}.{2SPACE}.{3SPACE,G
      REY 1,RVOFF}G"                              <133>
12016 PRINT SPC(9)"{GREEN,SPACE,GREY 1}TTT
      TTTTTTTTTTTTTTTT{ORANGE,SPACE}"             <169>
12019 RA=12:OG=1:UG=1:WE=0                        <160>
12020 PRINT"{GREEN}HIER STEHT EIN BAUM DER
       SELTENEN GATTUNG                           <098>
12030 PRINT"{UP}'ZYPRESSUS JACARANDA              <250>
12050 GOSUB 760                                   <200>
12120 PRINT"{WHITE,DOWN}MOEGLICHE RICHTUNG
      EN : N,S,O"                                 <187>
12125 GOSUB 750                                   <003>
12130 GOSUB 100                                   <136>
12135 IF S=5 THEN GOTO 12000                      <228>
12137 IF V=29 AND S=2 AND R=3 THEN GOSUB 1
      2400:R=0                                    <126>
12140 ON R GOTO 17000,11500,12160,13000,12
      160,12160                                   <115>
12150 GOTO 12200                                  <016>
12160 GOSUB 701                                   <246>
12200 IF V=30 AND WE=1 AND MA=1 THEN GOSUB
       12500                                      <006>
12299 GOSUB 777                                   <179>
12300 GOTO 12130                                  <072>
12400 PRINT"{DOWN}SIE STEHEN FUENF SCHRITT
      E WESTLICH DES                              <253>
12410 PRINT"BAUMES.":Q=1:WE=1:RETURN              <236>
12500 PRINT"{DOWN}SIE LEGEN EIN FALTBOOT F
      REI.":RA$(12,9)=G$(9):WL=1:GOSUB 760        <238>
12510 Q=1:RETURN                                  <021>
13000 PRINT SPC(9)"{ORANGE,SPACE,GREY 1}GG
      GGGGGGGGGGGGGGGGGG{ORANGE,SPACE}"           <090>
13001 PRINT SPC(9)"{GREY 1}R{4SPACE}G{15SP
      ACE}G"                                      <108>
13002 PRINT SPC(9)"R FPG{15SPACE,BROWN,SPA
      CE,GREY 1}G"                                <053>
13003 PRINT SPC(9)"RRM{2SPACE}TBFPG{2SPACE
      ,BROWN,2SPACE,GREY 1,6SPACE,BROWN,SP
      ACE,GREY 1}G"                               <150>
13004 PRINT SPC(9)"RR TBFB{2SPACE,BROWN}UB
      I{GREY 1}G{7SPACE,BROWN,SPACE,GREY 1
      }G"                                         <211>
13005 PRINT SPC(9)"R{ORANGE,SPACE,GREY 1}G
      {ORANGE,SPACE,GREY 1}G{ORANGE,2SPACE
      ,GREY 1}E{BROWN}/{YELLOW,SPACE,BROWN
      ,SPACE}/{GREY 1,SPACE}TEDCFRGGG"            <142>
13006 PRINT SPC(9)"R G{ORANGE,SPACE,GREY 1
      }I{ORANGE,2SPACE,BROWN}/{YELLOW,2SPA
      CE,BROWN}/{GREY 1}YDCFRPG{2SPACE}TG"        <083>
13007 PRINT SPC(9)"R{ORANGE,SPACE,GREY 1}B
      {ORANGE,SPACE,GREY 1}G{ORANGE,SPACE,
      BROWN}/{GREEN,2SPACE,BROWN}/{SPACE,G
      REY 1}G{ORANGE,6SPACE,GREY 1}YEDG"          <121>
13008 PRINT SPC(9)"R{ORANGE,SPACE,GREY 1}Y
      {ORANGE,SPACE,GREY 1}B{BROWN}/{GREEN
      ,2SPACE,BROWN}/{SPACE,PURPLE,SPACE,G
      REY 1}I{BROWN,SPACE,ORANGE,3SPACE,GR
      EY 1}I{ORANGE,4SPACE,GREY 1}G"              <147>
13009 PRINT SPC(9)"R{ORANGE,SPACE,GREY 1}R
      {ORANGE,SPACE,BROWN}/{GREEN,2SPACE,B
      ROWN}/{3SPACE,GREY 1}I{ORANGE,4SPACE
      ,GREY 1}I{ORANGE,4SPACE,GREY 1}G"           <176>
13010 PRINT SPC(9)"R{GREY 2}EB{BROWN}UDI/{
      ORANGE,2SPACE,BROWN,2SPACE,GREY 1}G{
      BROWN,SPACE,ORANGE,3SPACE,GREY 1}G{O
      RANGE,4SPACE,GREY 1}R"                      <240>
13011 PRINT SPC(9)"R{ORANGE,2SPACE,BROWN}J
      RK{GREY 2}EFPRFC*DEDCFRFC{GREY 1}G"         <114>
13012 PRINT SPC(9)"R{BROWN,SPACE,GREY 2}.{
      BROWN,2SPACE,GREY 2}.{BROWN,6SPACE,G
      REY 2}.{BROWN,3SPACE,GREY 2}.{3SPACE
      }.{GREY 1}G"                                <137>
13013 PRINT SPC(9)"R{BROWN,3SPACE,GREY 2}.
      {BROWN,3SPACE,GREY 2}.{BROWN,9SPACE,
      GREY 2}.{BROWN,2SPACE,GREY 1}G"             <022>
13014 PRINT SPC(9)"R{BROWN,SPACE,GREY 2}.{
      SPACE,BROWN,2SPACE,GREY 2}.{BROWN,4S
      PACE,GREY 2}.{BROWN,2SPACE,GREY 2}.{
      BROWN,SPACE,GREY 2,4SPACE}.{GREY 1}R
      "                                           <247>
13015 PRINT SPC(9)"R{BROWN,7SPACE,GREY 2}.
      {BROWN,7SPACE,GREY 2}.{BROWN,2SPACE,
      GREY 2}.{BROWN,SPACE,GREY 1}G"              <024>
13016 PRINT SPC(9)"{ORANGE,SPACE,GREY 1}TT
      TTTTTTTTTTTTTTTTTT{ORANGE,SPACE}"           <074>
13017 IF SH=0 THEN PRINT"{12UP}"                  <061>
13018 IF SH=0 THEN PRINT SPC(17)"{YELLOW}X
      {DOWN,2LEFT,GREEN}UK{DOWN,3LEFT}UK{D
      OWN,3LEFT}UK{9DOWN}"                        <234>
13019 RA=13:OG=1:UG=1                             <164>
13020 PRINT"{GREEN}SIE STEHEN VOR EINEM BA
      UMSTAMM,DER QUER                            <119>
13030 PRINT"UEBER EINER SCHLUCHT LIEGT.           <031>
13050 GOSUB 760                                   <182>
13120 PRINT"{WHITE,DOWN}MOEGLICHE RICHTUNG
      EN : W,O"                                   <105>
13125 GOSUB 750                                   <243>
13130 GOSUB 100                                   <120>
13135 IF S=5 THEN GOTO 13000                      <214>
13140 ON R GOTO 13160,13160,12000,13500,13
      160,13160                                   <251>
13150 GOTO 13200                                  <016>
13160 GOSUB 701                                   <230>
13200 IF SH=0 AND V=2 AND S=11 AND RA$(13,
      5)=G$(5)THEN GOSUB 13400:Q=1                <171>
13210 IF SH=0 AND V=20 THEN GOTO 13600            <111>
13299 GOSUB 777                                   <161>
13300 GOTO 13130                                  <070>
13400 PRINT"{DOWN}DIE SCHLANGE FRISST VON
      DEN FRUECHTEN.{DOWN}                        <081>
13410 PRINT"SIE KRUEMMT SICH,VERLIERT DEN
      HALT UND                                    <154>
13420 PRINT"FAELLT IN DIE SCHLUCHT!":SH=1:
      RA$(13,5)="":RETURN                         <127>
13500 IF SH=1 THEN GOTO 14000                     <042>
13510 GOTO 13600                                  <250>
13600 PRINT"{DOWN}DIE AUF DEM BAUMSTAMM LI
      EGENDE SCHLANGE                             <184>
13601 PRINT"HAT SIE GEBISSEN!!                    <190>
13605 ZA=1:LI=2000:D=TI:GOSUB 100                 <041>
13610 IF V=32 THEN GOSUB 777:PRINT"{DOWN}S
      IE HABEN KEINE SPRITZE!":GOTO 13605         <192>
13620 GOTO 900                                    <150>
14000 PRINT SPC(9)"{GREY 1,SPACE}GGGGGGGGG
      GGGGGGGGGG "                                <130>
```

```
14001 PRINT SPC(9)"R{GREEN,8SPACE,WHITE,RV
      SON}£{2SPACE}*{GREEN,RVOFF,8SPACE,GR
      EY 1}G"                                <160>
14002 PRINT SPC(9)"R{GREEN,3SPACE,GREY 1}/
      M{GREEN,2SPACE,GREY 2}N{WHITE,SPACE}
      TT{GREEN,SPACE,GREY 2}M{GREEN,2SPACE
      ,GREY 2,SPACE,GREEN,3SPACE,GREY 1}/G
      "                                      <002>
14003 PRINT SPC(9)"R{GREEN,2SPACE,GREY 1}/
      {GREEN,2SPACE,GREY 1}M{GREY 2}N{GREE
      N,6SPACE,GREY 2}MMM{GREEN,2SPACE,GRE
      Y 1}/{GREEN,SPACE,GREY 1}G"            <018>
14004 PRINT SPC(9)"R{GREEN,SPACE,GREY 1}/{
      GREEN,3SPACE,GREY 2}N{GREEN,7SPACE,G
      REY 2}N{SPACE,GREEN,SPACE,GREY 2}M{G
      REY 1}/{GREEN,2SPACE,GREY 1}G"         <241>
14005 PRINT SPC(9)"R/{GREEN,3SPACE,GREY 2}
      N{GREEN,7SPACE,GREY 2}N{GREEN,4SPACE
      ,GREY 2}M{GREEN,2SPACE,GREY 1}R"       <064>
14006 PRINT SPC(9)"R{GREEN}CFRTRFCFRPG{GRE
      Y 2}N{GREEN,SPACE}GPRCDCR{GREY 1}R"    <056>
14007 PRINT SPC(9)"R{GREEN,5SPACE}BFDT{2SP
      ACE}TYTECRP{2SPACE,GREY 1}R"           <186>
14008 PRINT SPC(9)"R{GREEN}GPIOIOPGGPGPOI{
      RVSON}UUYU{RVOFF}TO{GREY 1}R"          <238>
14009 PRINT SPC(9)"R{GREEN,7SPACE}T{SPACE,
      ORANGE,SPACE}NM{2SPACE,GREEN,6SPACE,
      GREY 1}R"                              <054>
14010 PRINT SPC(9)"R{GREEN,6SPACE,ORANGE,2
      SPACE,GREEN,SPACE,ORANGE}N{2SPACE}MF
      PPPPPP{GREY 1}G"                       <017>
14011 PRINT SPC(9)"R{GREEN,8SPACE,ORANGE}N
      {GREEN,5SPACE,ORANGE}&{GREEN,2SPACE,
      ORANGE}&{GREEN,SPACE,ORANGE}&{GREY 1
      }G"                                    <146>
14012 PRINT SPC(9)"R{GREEN,7SPACE,ORANGE}N
      {GREEN,2SPACE,ORANGE}& &{GREEN,3SPAC
      E,ORANGE}&{GREEN,SPACE,ORANGE}&{GREE
      N,SPACE,GREY 1}G"                      <218>
14013 PRINT SPC(9)"R{GREEN,5SPACE,ORANGE,S
      PACE}N{GREEN,SPACE,ORANGE,RVSON}B{GR
      EEN,RVOFF,3SPACE,WHITE,SPACE,ORANGE,
      RVSON}B{WHITE,RVOFF,SPACE,ORANGE}&{G
      REEN,2SPACE,ORANGE,SPACE}&{GREY 1}G"   <132>
14014 PRINT SPC(9)"R{GREEN,4SPACE,ORANGE,S
      PACE}N{GREEN,3SPACE,ORANGE,RVSON}B{R
      VOFF,SPACE,RVSON}B{RVOFF,4SPACE}PPPP
      {GREY 1}G"                             <030>
14015 PRINT SPC(9)"R{GREEN,4SPACE,ORANGE}N
      {GREEN,3SPACE,ORANGE,3SPACE,GREEN,2S
      PACE,ORANGE}&{GREEN,2SPACE,ORANGE}M{
      GREEN,3SPACE,GREY 1}G"                 <069>
14016 PRINT SPC(9)" TTTTTTTTTTTTTTTTTTTT{G
      REEN,SPACE}"                           <090>
14019 RA=14:OG=27:UG=27                      <217>
14020 PRINT"{GREEN}HIER TEILT SICH DER WEG
      .EIN BREITER WEG                       <132>
14030 PRINT"MIT VIELEN FUSSABDRUECKEN FUEH
      RT NACH                                <146>
14040 PRINT"SUEDEN,EIN SCHMALER PFAD NACH
      OSTEN.                                 <005>
14050 GOSUB 760                              <166>
14120 PRINT"{WHITE,DOWN}MOEGLICHE RICHTUNG
      EN : S,W,O"                            <239>
14125 GOSUB 750                              <227>
14130 GOSUB 100                              <104>
14135 IF S=5 THEN GOTO 14000                 <200>
14140 ON R GOTO 14160,15000,13000,14500,14
      160,14160                              <001>
14150 GOTO 14200                             <016>
14160 GOSUB 701                              <214>
14200 IF V=6 AND S=27 THEN GOSUB 14400       <047>
14210 IF V=25 AND G=2 OR V=25 AND S=28 THE
      N GOSUB 14450                          <082>
14299 GOSUB 777                              <145>
14300 GOTO 14130                             <070>
14400 PRINT"{DOWN}AUF DEM BODEN LIEGT EIN
      GLIMMENDES STUE-CK HOLZ!":Q=1:RETURN   <241>
14450 IF SP$(2)=""THEN PRINT"{DOWN}SIE HAB
      EN DAS NICHT!":Q=1:RETURN              <004>
14460 GOSUB 705:PRINT"DIE ZUENDSCHNUR BREN
      NT!"                                   <146>
14465 IF PA=0 THEN PRINT"{DOWN}ES IST ABER
       KEIN PULVER DARIN."                   <174>
14470 IF PA=1 THEN Q=1:ZA=1:D=TI:LI=3600:Z
      U=1                                    <090>
14480 RETURN                                 <058>
14500 RA=19:OG=1:UG=1                        <251>
14520 PRINT"{GREEN,2DOWN}SIE STEHEN AUF EI
      NER WALDLICHTUNG.                      <139>
14550 GOSUB 760                              <158>
14600 PRINT"{WHITE,DOWN}MOEGLICHE RICHTUNG
      EN : S,W{DOWN}"                        <020>
14606 GOSUB 750                              <200>
14607 IF LE=0 THEN GOTO 14950                <119>
14608 IF LE=1 THEN PRINT"{DOWN}AUF DEM BOD
      EN LIEGT EIN TOTER JAGUAR."            <225>
14610 GOSUB 100                              <076>
14620 ON R GOTO 14640,16000,14000,14640,14
      640,14640                              <045>
14625 IF S=5 THEN GOTO 14520                 <218>
14630 GOTO 14849                             <058>
14640 GOSUB 701                              <186>
14849 GOSUB 777                              <189>
14850 GOTO 14610                             <146>
14950 PRINT"{DOWN}EIN JAGUAR TAUCHT AUF DI
      E LICHTUNG AUF.                        <054>
14951 PRINT"{2DOWN}ER GREIFT SIE AN!!":ZA=
      1:D=TI:LI=3600:LE=1                    <143>
14952 GOSUB 100                              <164>
14953 IF V=20 AND S=29 AND SP$(1)<>""AND G
      =1 THEN 14960                          <076>
14954 RA=14:GOTO 900                         <232>
14960 PRINT"{DOWN}GESCHAFFT!ABER DER JAGUA
      R HAT IHNEN DIE                        <075>
14961 PRINT"{DOWN}PULSADERN DER LINKEN HAN
      D AUFGERISSEN!"                        <000>
14962 PRINT"{DOWN}SIE DROHEN ZU VERBLUTEN!
      !":D=TI                                <095>
14963 GOSUB 100                              <175>
14964 IF V=26 AND G=6 AND SP$(6)<>""THEN G
      OSUB 705:GOTO 14970                    <044>
14965 GOSUB 777:FOR I=0 TO 3000:NEXT         <174>
14966 RA=16:GOTO 900                         <116>
14970 PRINT"{DOWN}DIE BLUTUNG KOMMT ZUM ST
      ILLSTAND.SIE{4SPACE}SIND GERETTET!":
      ZA=0:GOTO 14610                        <084>
15000 PRINT SPC(9)"{YELLOW,SPACE,GREY 1}GG
      GGGGGGGGGGGGGGGGGG "                   <085>
15001 PRINT SPC(9)"R{GREEN,20SPACE,GREY 1}
      G"                                     <175>
15002 PRINT SPC(9)"R{GREEN,20SPACE,GREY 1}
      G"                                     <176>
15003 PRINT SPC(9)"R{GREEN,20SPACE,GREY 1}
      G"                                     <177>
15004 PRINT SPC(9)"R{GREEN,20SPACE,GREY 1}
      G"                                     <178>
15005 PRINT SPC(9)"R{GREEN,20SPACE,GREY 1}
      G"                                     <179>
15006 PRINT SPC(9)"R{GREEN,3SPACE,GREY 2}A
      AA{YELLOW}NM{GREY 2}AAAAAA{YELLOW}NM
      {GREY 2}AAA{GREEN,SPACE,GREY 1}G"      <087>
15007 PRINT SPC(9)"R{GREEN,2SPACE,GREY 2}A
      BB{YELLOW}N{SPACE,GREEN,SPACE,YELLOW
      }M{GREY 2}BBBB{YELLOW}N{GREEN,2SPACE
      ,YELLOW}M{GREY 2}BBA{GREY 1}G"         <239>
15008 PRINT SPC(9)"R{GREEN,SPACE,GREY 2}AB
      {GREEN,SPACE,YELLOW}N{3SPACE,GREEN,S
      PACE,YELLOW}M{GREEN,2SPACE,YELLOW,SP
      ACE}PEEO{GREEN,2SPACE,GREY 2}B{GREY
      1}G"                                   <026>
15009 PRINT SPC(9)"R{GREY 2}ABB{GREEN,SPAC
      E,YELLOW}PEDDEO{GREEN,3SPACE,YELLOW}
      WUIR{GREEN,3SPACE,GREY 1}G"            <121>
15010 PRINT SPC(9)"R{GREY 2}BB{GREEN,2SPAC
      E,YELLOW}R{GREEN,SPACE,YELLOW}UI{GRE
      EN,SPACE,YELLOW}R{GREEN,2SPACE,BROWN
      ,2SPACE,YELLOW}TT{GREEN,4SPACE,GREY
      1}G"                                   <066>
15011 PRINT SPC(9)"R{GREY 2}B{GREEN,2SPACE
      }I{YELLOW}R{GREEN,SPACE,YELLOW}BB{GR
      EEN,SPACE,YELLOW}R{GREEN}K{SPACE,BRO
      WN}&{GREY 3,SPACE,GREEN,3SPACE}I K{G
      REY 1}G"                               <206>
15012 PRINT SPC(9)"R{GREY 2}B{GREEN,SPACE}
      U{2SPACE,YELLOW}TEET{GREEN,2SPACE,BR
      OWN}U+K{GREEN,SPACE}J I{2SPACE,GREY
      1}G"                                   <093>
15013 PRINT SPC(9)"R{GREEN,4SPACE}K{2SPACE
      }I J{SPACE,BROWN}DEF{GREEN}I{3SPACE}
      J{SPACE,GREY 1}G"                      <102>
15014 PRINT SPC(9)"R{GREEN,SPACE}K{3SPACE}
```

**Listing »Odyssey« (Fortsetzung)**

```
      U{3SPACE}I{2SPACE}U{3SPACE}K{3SPACE,
      GREY 1}G"                              <209>
15015 PRINT SPC(9)"H{20SPACE}G"              <178>
15016 PRINT SPC(9)"{GREEN,SPACE,GREY 1}TTT
      TTTTTTTTTTTTTTTTT{GREEN,SPACE}"        <012>
15019 RA=15                                  <016>
15020 PRINT"{GREEN}SIE STEHEN VOR DEM EING
      ANG DES KANNIBA-                       <179>
15030 PRINT"LENDORFES.                       <242>
15050 GOSUB 760                              <150>
15120 PRINT"{WHITE,DOWN}MOEGLICHE RICHTUNG
      EN : N{DOWN}"                          <054>
15125 GOSUB 750                              <211>
15127 IF KA=0 THEN PRINT"{DOWN}DIE KANNIBA
      LEN HABEN SIE ENTDECKT UND{3SPACE}GR
      EIFEN SIE AN!"                         <135>
15129 Q=1:KA=1:ZA=1:D=TI:LI=3600             <209>
15130 GOSUB 100                              <088>
15135 IF S=5 THEN GOTO 15000                 <186>
15140 ON R GOTO 14000,15160,15160,15160,15
      160,15160                              <221>
15150 GOTO 15300                             <048>
15160 GOSUB 701                              <198>
15300 GOTO 15130                             <070>
16000 PRINT SPC(9)"{GREEN,SPACE,GREY 1}EEE
      EEEEEEEEEEEEEEEEE{GREEN,SPACE}"        <012>
16001 PRINT SPC(9)"{GREY 1}H{GREEN,20SPACE
      ,GREY 1}G"                             <051>
16002 PRINT SPC(9)"H{GREEN,20SPACE,GREY 1}
      G"                                     <160>
16003 PRINT SPC(9)"H{GREEN,8SPACE,YELLOW,1
      0SPACE,GREEN,2SPACE,GREY 1}H"          <203>
16004 PRINT SPC(9)"H{GREEN,8SPACE,YELLOW,S
      PACE,RVSON}£{4SPACE,RVOFF}£M{GREEN,4
      SPACE,GREY 1}G"                        <232>
16005 PRINT SPC(9)"H{GREEN,7SPACE,YELLOW,S
      PACE,RVSON}£TTTTT*{GREEN,RVOFF,SPACE
      ,YELLOW}M{GREEN,3SPACE,GREY 1}G"       <081>
16006 PRINT SPC(9)"H{GREEN,6SPACE,YELLOW,S
      PACE,RVSON}£ZZH HZZ*{GREEN,RVOFF,SPA
      CE,YELLOW}M{GREEN,2SPACE,GREY 1}G"     <110>
16007 PRINT SPC(9)"H{GREEN,5SPACE,ORANGE,S
      PACE,YELLOW,RVSON}£{4SPACE}H{4SPACE}
      *{RVOFF}M{GREEN,2SPACE,GREY 1}G"       <103>
16008 PRINT SPC(9)"H{GREY 2}AAAAAA{ORANGE}
      PPPPPPPPPP G{GREY 2}AA{GREY 1}G"       <221>
16009 PRINT SPC(9)"H{GREY 2}BBBBBB{ORANGE}
      NNNNNNNNNN{GREEN,SPACE,ORANGE}G{GREY
       2}HH{GREY 1}G"                        <106>
16010 PRINT SPC(9)"H{GREY 2}BBBBBB{ORANGE}
      NNNNNNNNNN{GREEN,SPACE,ORANGE}G{GREY
       2}HH{GREY 1}G"                        <077>
16011 PRINT SPC(9)"H{ORANGE}U{GREEN,SPACE,
      ORANGE}J{2SPACE}INNNNNNNNNNN{GREEN,S
      PACE,ORANGE}J{GREEN,SPACE,GREY 1}H"    <250>
16012 PRINT SPC(9)"H{GREEN,3SPACE,ORANGE}I
      {GREEN,3SPACE,ORANGE}TTTTTTTTT{GREEN
      ,3SPACE,ORANGE}J{GREY 1}G"             <223>
16013 PRINT SPC(9)"H{GREEN,5SPACE,ORANGE}I
      {GREEN,SPACE,ORANGE}K{GREEN,2SPACE,O
      RANGE}I{GREEN,SPACE,ORANGE}J{GREEN,2
      SPACE,ORANGE}U J{GREEN,2SPACE,GREY 1
      }G"                                    <137>
16014 PRINT SPC(9)"H{GREEN,SPACE,ORANGE}U{
      GREEN,SPACE,ORANGE}K{GREEN,4SPACE,OR
      ANGE}U{GREEN,3SPACE,ORANGE,SPACE}K{G
      REEN,4SPACE,ORANGE}I{GREEN,SPACE,GRE
      Y 1}G"                                 <235>
16015 PRINT SPC(9)"H{GREEN,3SPACE,ORANGE}I
      {GREEN,2SPACE,ORANGE}U{GREEN,4SPACE,
      ORANGE}J{GREEN,3SPACE,ORANGE}J{GREEN
      ,4SPACE,GREY 1}H"                      <125>
16016 PRINT SPC(9)"{GREEN,SPACE,GREY 1}TTT
      TTTTTTTTTTTTTTTTT{GREEN,SPACE}"        <252>
16019 RA=16                                  <064>
16020 PRINT"{GREEN}SIE BEFINDEN SICH AN DE
      R RUECKSEITE DES                       <011>
16030 PRINT"DORFES.HIER STEHT DIE OPFERHUE
      TTE DER                                <067>
16040 PRINT"KANNIBALEN.SIE IST ANSCHEINEND
       UNBEWACHT                             <109>
16050 GOSUB 760                              <134>
16120 PRINT"{WHITE,DOWN}MOEGLICHE RICHTUNG
      EN : N{DOWN}"                          <036>
16125 GOSUB 750                              <193>
16130 GOSUB 100                              <072>
16135 IF S=5 THEN GOTO 16000                 <172>
16140 ON R GOTO 14500,16160,16160,16160,16
      160,16160                              <180>
16150 GOTO 16300                             <048>
16160 GOSUB 701                              <202>
16300 IF V=28 AND S=30 THEN GOSUB 705:GOSU
      B 16800                                <107>
16310 IF V=27 AND S=30 THEN GOSUB 705:GOSU
      B 16700                                <019>
16499 GOSUB 777                              <059>
16500 GOTO 16130                             <016>
16700 PRINT"{DOWN}IN DER HUETTE IST EIN GE
      FESSELTER WEIS-                        <043>
16701 PRINT"SER MANN.ER SAGT ZU IHNEN:       <124>
16702 PRINT"{DOWN}"CHR$(34)"WER SIE AUCH I
      MMER SIND,FLIEHEN SIE!                 <028>
16703 PRINT"DIE HUETTE IST BEWACHT.GEHEN S
      IE ZUM                                 <154>
16704 PRINT"BAUM NOERDLICH VOM FUSS DER TR
      EPPE!FUENF                             <179>
16705 PRINT"{UP}FUSS WESTLICH VON IHM LIEG
      T MEIN FALT-                           <055>
16706 PRINT"BOOT VERGRABEN.FAHREN SIE ZUR
      FELSEN-                                <035>
16707 PRINT"INSEL UND HOLEN SIE HILFE!"CHR
      $(34)                                  <126>
16710 Q=1:MA=1:RETURN                        <095>
16800 PRINT"{DOWN}DIE KANNIBALEN HABEN SIE
       ENTDECKT UND{3SPACE}GREIFEN SIE AN!
      "                                      <252>
16810 Q=1:KA=1:ZA=1:D=TI:LI=3600:RETURN      <226>
17000 PRINT SPC(9)"{GREEN,SPACE,GREY 1}EEE
      EEEEEEEEEEEEEEEEE{GREEN,SPACE}"        <252>
17001 PRINT SPC(9)"{GREY 1}H{GREEN,20SPACE
      ,GREY 1}G"                             <035>
17002 PRINT SPC(9)"H{SPACE,GREEN,19SPACE,G
      REY 1}G"                               <018>
17003 PRINT SPC(9)"H{GREEN,20SPACE,GREY 1}
      G"                                     <145>
17004 PRINT SPC(9)"H{GREEN,7SPACE,GREY 1}*
      {2SPACE,GREEN,10SPACE,GREY 1}G"        <050>
17005 PRINT SPC(9)"H{GREEN,3SPACE,GREY 1,S
      PACE}**{RVSON}£ *£*{GREEN,RVOFF,9SPA
      CE,GREY 1}G"                           <053>
17006 PRINT SPC(9)"H{WHITE,3SPACE,GREY 1,R
      VSON}£ M{3SPACE}M *{WHITE,RVOFF,8SPA
      CE,GREY 1}G"                           <113>
17007 PRINT SPC(9)"H{LIG.BLUE,RVSON,20SPAC
      E,GREY 1,RVOFF}G"                      <113>
17008 PRINT SPC(9)"H{LIG.BLUE,RVSON,20SPAC
      E,GREY 1,RVOFF}G"                      <240>
17009 PRINT SPC(9)"H{LIG.BLUE,RVSON,20SPAC
      E,GREY 1,RVOFF}G"                      <115>
17010 PRINT SPC(9)"H{LIG.BLUE,RVSON}EE{RVO
      FF}UT Y{RVSON}TPE POT{RVOFF}YT T{GRE
      EN,SPACE,LIG.BLUE}TY{GREY 1}G"         <220>
17011 PRINT SPC(9)"H{GREEN,2SPACE,GREY 2}.
      {GREEN,2SPACE,GREY 2}.{GREEN,4SPACE,
      GREY 3,2SPACE,GREY 2}.{GREY 3,SPACE,
      GREEN,2SPACE,GREY 2}.{GREEN,3SPACE,G
      REY 1}G"                               <155>
17012 PRINT SPC(9)"H{GREY 2}.{GREEN,2SPACE
      ,GREY 2}.{GREEN,3SPACE,GREY 2}.{GREE
      N,6SPACE,GREY 2}.{GREEN,2SPACE,GREY
      2}.{GREEN,SPACE,GREY 2}.{GREY 1}H"     <146>
17013 PRINT SPC(9)"H{GREEN,10SPACE,GREY 2}
      .{GREEN,9SPACE,GREY 1}G"               <213>
17014 PRINT SPC(9)"H{GREEN,SPACE,GREY 2}.{
      GREEN,SPACE,GREY 2,SPACE}.{GREEN,SPA
      CE,GREY 2,SPACE,GREEN,SPACE,GREY 2}.
      {GREEN,3SPACE,GREY 2}.{GREEN,2SPACE,
      GREY 2}.{GREEN,2SPACE,GREY 2}.{GREEN
      ,SPACE,GREY 1}H"                       <038>
17015 PRINT SPC(9)"H{GREEN,20SPACE,GREY 1}
      H"                                     <091>
17016 PRINT SPC(9)"{GREEN,SPACE,GREY 1}TTT
      TTTTTTTTTTTTTTTTT{GREEN,SPACE}"        <236>
17019 RA=17                                  <112>
17020 PRINT"{GREEN}SIE STEHEN AN DER NORDK
      UESTE DER INSEL.                       <034>
17030 PRINT"IN DER FERNE IST EINE FELSENIN
      SEL ZU                                 <206>
17040 PRINT"ERKENNEN.                        <240>
17050 GOSUB 760                              <118>
17120 PRINT"{WHITE,DOWN}MOEGLICHE RICHTUNG
      EN : S{DOWN}"                          <040>
```

```
17125 GOSUB 750                                    <177>
17130 GOSUB 100                                    <054>
17135 IF S=5 THEN GOTO 17000                       <156>
17140 ON R GOTO 17160,12000,17160,17160,17
      160,17160                                    <218>
17150 GOTO 17200                                   <014>
17160 GOSUB 701                                    <166>
17200 IF S=31 AND SP$(9)<>""THEN GOSUB 705
      :WA=1                                        <241>
17210 IF WA=1 AND V=31 AND S=32 THEN 17310         <081>
17300 GOTO 17130                                   <070>
17310 DD=1:PRINT"{CLR}"SPC(16)"{WHITE}DAS
      WAR"                                         <170>
17320 PRINT SPC(13)"{10DOWN,YELLOW}O D Y S
       S E E                                       <125>
17330 PRINT SPC(14)"{3DOWN,YELLOW}T E I L{
      3SPACE}1                                     <200>
17340 FOR I=0 TO 6000:NEXT:PRINT"{CLR}"SPC
      (10)"{WHITE}ERLEBEN SIE DEMNAECHST           <229>
17420 FOR X=0 TO 2:GOSUB 20060:GOTO 20004          <022>
17430 PRINT SPC(14)"{YELLOW,4DOWN}T E I L{
      2SPACE}2":GOSUB 750:PRINT"{WHITE,CLR
      }":END                                       <221>
20000 READ S,B:IF S<>-1 THEN POKE 55272+S,
      F:POKE 1000+S,B:FOR I=0 TO 200:NEXT:
      GOTO 20000                                   <015>
20001 IF F=2 THEN F=1:GOTO 20000                   <095>
20002 FOR X=0 TO 2:RESTORE:FOR I=0 TO 24:R
      EAD S,B:IF S=-1 THEN NEXT I                  <080>
20003 POKE 1000+S,128+B:POKE 1000+S,B:NEXT
       I:NEXT::GOSUB 20050:FOR X=0 TO 2:GO
      SUB 20060                                    <003>
20004 PRINT SPC(4-X)"{UP,YELLOW,2SPACE}Ḡ{S
      PACE,GREEN,2SPACE,YELLOW}ḠḠ{GREEN,2S
      PACE,YELLOW,SPACE}Ḡ Ḡ{2SPACE}ḠḠḠ{GRE
      EN,2SPACE,YELLOW}ḠḠḠ{GREEN,2SPACE,YE
      LLOW}ḠḠḠ{GREEN,2SPACE,YELLOW}Ḡ{GREEN
      ,SPACE,YELLOW}Ḡ{GREEN,SPACE}"                <082>
20005 GOSUB 20050                                  <246>
20006 PRINT SPC(X)"{YELLOW,SPACE}//M̲ /{GRE
      EN,SPACE,YELLOW}/M̲{GREEN,SPACE,YELLO
      W}////Ṟ/{GREEN,2SPACE,YELLOW}/Ḡ/{GRE
      EN,2SPACE,YELLOW}/Ḡ/{GREEN,2SPACE,YE
      LLOW}/Ḡ////Ṟ"                                <201>
20007 GOSUB 20050                                  <248>
20008 PRINT SPC(4-X)"{RVSON}£ ¥{RVOFF}/Ḡ{R
      VSON,2SPACE}¥{RVOFF}/Ḡ{RVSON,SPACE,R
      VOFF,SPACE,RVSON,SPACE,RVOFF,SPACE}Ḡ
      {RVSON,3SPACE,RVOFF}/{SPACE,RVSON,3S
      PACE,RVOFF}/{SPACE,RVSON,3SPACE,RVOF
      F}/{SPACE,RVSON,SPACE,RVOFF,SPACE,RV
      SON,SPACE,RVOFF,SPACE}Ḡ"                     <228>
20009 GOSUB 20050                                  <250>
20010 PRINT SPC(X)"{RVSON,SPACE,RVOFF,SPAC
      E,YELLOW,RVSON,SPACE,GREEN,RVOFF,SPA
      CE,YELLOW}Ṟ{RVSON,SPACE,RVOFF,SPACE,
      YELLOW,RVSON,SPACE,GREEN,RVOFF,SPACE
      ,YELLOW}Ḡ{RVSON,SPACE,RVOFF,SPACE,RV
      SON,SPACE,RVOFF,SPACE}Ḡ{RVSON,SPACE,
      RVOFF,SPACE}LF{SPACE,RVSON,SPACE,RVO
      FF,SPACE}LF{SPACE,RVSON,SPACE,GREEN,
      RVOFF,SPACE,YELLOW}LF{SPACE,RVSON,SP
      ACE,RVOFF,SPACE,RVSON,SPACE,RVOFF,SP
      ACE}Ḡ                                        <149>
20011 GOSUB 20050                                  <252>
20012 PRINT SPC(4-X)"{RVSON,SPACE,RVOFF,SP
      ACE,RVSON,SPACE,RVOFF,SPACE}Ṟ{RVSON,
      SPACE,RVOFF,SPACE,RVSON,SPACE,RVOFF,
      SPACE}Ḡ{RVSON,SPACE,RVOFF}/{RVSON,SP
      ACE,RVOFF}/{SPACE,RVSON,SPACE,RVOFF}
      / /Ṟ{RVSON,SPACE,RVOFF}/ /Ḡ{RVSON,SP
      ACE,RVOFF}/ /Ḡ{RVSON,SPACE,RVOFF}/{R
      VSON,SPACE,RVOFF}/ "                         <139>
20013 GOSUB 20050                                  <254>
20014 PRINT SPC(X)"{RVSON,SPACE,RVOFF,SPAC
      E,RVSON,SPACE,RVOFF,SPACE}Ḡ{RVSON,SP
      ACE,RVOFF,SPACE,RVSON,SPACE,RVOFF,SP
      ACE}Ḡ¥{RVSON,SPACE,RVOFF}£Ḡ{SPACE,RV
      SON,3SPACE,RVOFF,SPACE}Ṟ{RVSON,3SPAC
      E,RVOFF,SPACE}Ṟ{RVSON,3SPACE,RVOFF}/
      ¥{RVSON,SPACE,RVOFF}£Ḡ "                      <136>
20015 GOSUB 20050                                  <000>
20016 PRINT SPC(4-X)"{RVSON,SPACE,RVOFF}/{
      RVSON,SPACE,GREEN,RVOFF,SPACE,YELLOW
      }Ḡ{RVSON,SPACE,RVOFF}/{RVSON,SPACE,R
      VOFF,SPACE}Ḡ{SPACE,RVSON,SPACE,RVOFF
      ,SPACE}Ṟ /{GREEN,SPACE,YELLOW,RVSON,
      SPACE,GREEN,RVOFF,SPACE,YELLOW}Ṟ/{GR
      EEN,SPACE,YELLOW,RVSON,SPACE,GREEN,R
      VOFF,SPACE,YELLOW}Ḡ{RVSON,SPACE,RVOF
      F}/ /Ḡ{GREEN,SPACE,YELLOW,RVSON,SPAC
      E,RVOFF,SPACE}Ḡ                              <183>
20017 GOSUB 20050                                  <002>
20018 PRINT SPC(X)"{RVSON,3SPACE,RVOFF}/{S
      PACE,RVSON,3SPACE,RVOFF}/{2SPACE,RVS
      ON,SPACE,RVOFF,SPACE}Ḡ{SPACE,RVSON,3
      SPACE,RVOFF}/{SPACE,RVSON,3SPACE,RVO
      FF}/{SPACE,RVSON,3SPACE,RVOFF}/{2SPA
      CE,RVSON,SPACE,RVOFF,SPACE}Ḡ "               <100>
20019 GOSUB 20050                                  <004>
20020 PRINT SPC(4-X)"¥{RVSON,SPACE,RVOFF}£
      {2SPACE,RVSON,2SPACE,RVOFF}£{3SPACE,
      RVSON,SPACE,RVOFF}/{2SPACE,RVSON,2SP
      ACE,RVOFF}£{2SPACE,RVSON,2SPACE,RVOF
      F}£{2SPACE,RVSON,2SPACE,RVOFF}£{3SPA
      CE,RVSON,SPACE,RVOFF}/{2SPACE}"              <063>
20021 GOSUB 20050                                  <006>
20030 GOSUB 20060:PRINT"{DOWN,2SPACE,DOWN,
      2LEFT,2SPACE,DOWN,2LEFT,2SPACE,DOWN,
      2LEFT,2SPACE}";                              <114>
20031 PRINT"{DOWN,2LEFT,2SPACE,DOWN,2LEFT,
      2SPACE,DOWN,2LEFT,2SPACE,DOWN,2LEFT,
      2SPACE,DOWN,2LEFT}"                          <016>
20035 GOSUB 20060:PRINT"{DOWN,3LEFT,2SPACE
      ,DOWN,2LEFT,2SPACE,DOWN,2LEFT,2SPACE
      }";                                          <063>
20036 PRINT"{DOWN,2LEFT,2SPACE,DOWN,2LEFT,
      2SPACE,DOWN,3LEFT,2SPACE,DOWN,2LEFT,
      2SPACE,DOWN,2LEFT,2SPACE}"                   <243>
20040 NEXT                                         <238>
20049 GOTO 20065                                   <212>
20050 FOR I=0 TO 400:NEXT:RETURN                   <178>
20060 PRINT"{HOME,7DOWN}":RETURN                   <153>
20065 IF DD=1 THEN GOTO 17430                      <167>
20070 PRINT SPC(10)"{BLUE,2DOWN}EIN":GOSUB
       20050                                       <077>
20075 PRINT SPC(14)"{UP}ABENTEUERSPIEL":GO
      SUB 20050                                    <176>
20080 PRINT SPC(18)"{DOWN}VON":GOSUB 20050         <082>
20090 PRINT SPC(9)"{LIG.BLUE,DOWN}HANS":GO
      SUB 20050                                    <033>
20091 PRINT SPC(13)"{UP}-GUENTHER":GOSUB 2
      0050                                         <167>
20092 PRINT SPC(23)"{UP}SCHRECK":GOSUB 200
      50                                           <042>
20100 DATA 33,24,35,20,37,3,39,45,41,19,43
      ,15,45,6,47,20,49,23,51,1,53,18              <070>
20101 DATA 55,5,-1,-1,118,16,119,18,120,1,
      121,5,122,19,123,5,124,14,125,20             <086>
20102 DATA 126,9,127,5,128,18,129,20,-1,-1         <221>
20103 RETURN                                       <093>
25000 PRINT"{WHITE}SIE ERINNERN SICH:              <061>
25001 PRINT"{DOWN}SIE WAREN PASSAGIER AN B
      ORD DES LUXUS-                               <172>
25002 PRINT"LINERS 'QUEEN MARY' GEWESEN.DE
      R TAIFUN                                     <133>
25003 PRINT"HATTE DIE BESATZUNG VOLLKOMMEN
       UEBER-{3SPACE}RASCHT.                       <149>
25004 PRINT"DAS EINZIGE BRAUCHBARE RETTUNG
      SBOOT WAR                                    <207>
25005 PRINT"KURZ NACH DEM AUSSETZEN GEKENT
      ERT.                                         <125>
25006 PRINT"WAS DANN GESCHAH, WISSEN SIE N
      ICHT MEHR.                                   <167>
25007 PRINT"JETZT ERWACHEN SIE LANGSAM WIE
       DER AUS                                     <107>
25008 PRINT"IHRER BEWUSSTLOSIGKEIT.....            <250>
25009 :GOSUB 750:PRINT"{CLR}"CHR$(142):RET
      URN                                          <012>
30000 PRINT"{CLR}"SPC(17)"{WHITE,SPACE,RVS
      ON,SPACE,RVOFF}"                             <131>
30002 PRINT SPC(17)"{RVSON,3SPACE,RVOFF}"          <202>
30003 PRINT SPC(17)"{SPACE,RVSON,SPACE,RVO
      FF}"                                         <203>
30004 PRINT SPC(17)"{SPACE,RVSON,SPACE,RVO
      FF,DOWN}"                                    <153>
30005 RETURN                                       <089>
30100 PRINT"{DOWN}NOCH EIN VERSUCH? (J/N)"         <021>
30110 GET B$:IF B$=CHR$(74)THEN CLR:GOTO 1
      0                                            <109>
30120 IF B$=CHR$(78)THEN END                       <255>
30130 GOTO 30110                                   <232>
```

**Listing »Odyssey« (Schluß)**

# Der kleine Hobbit

**Erforschen Sie das Land der Fantasie. Finden Sie einen Schatz im Land der Elfen und Magier. Entrinnen Sie den bösen Mächten.**

Der kleine Hobbit ist im Prinzip nichts weiter als die deutsche Version eines der meistverkauften englischen Adventures überhaupt (sechsstellige Verkaufszahlen!): »The Hobbit« von Melbourn House. Bereits im ersten Adventurekurs (Sonderheft 2/85) habe ich Ihnen dieses Spiel, insbesondere seinen Parser, beschrieben. In der Tat, die Spectrum-Version von »The Hobbit« machte dieses Programm aufgrund seines ausgezeichneten Parsers (der lange als Standard bei Adventure-Tests betrachtet wurde) berühmt. Das folgende Listing ist eine deutsche Grafikversion des Hobbits von J.R.R. Tolkien. Natürlich wurde das englische Programm nicht einfach ins Deutsche übersetzt und die Handlung 1:1 übernommen. Ich habe den »kleinen Hobbit« vielmehr programmiert, nachdem ich sämtliche Tolkien-Bücher studiert habe. Die deutsche Hobbit-Version ähnelt nicht zuletzt deshalb viel mehr der Handlung des Buches als die englische es tut. Dieses Programm soll keineswegs eine Konkurrenz zum englischen Hobbit sein, sondern vielmehr ein kleines Tolkien-Adventure für alle, die kein Englisch können. Der Parser des Spiels ist ausgezeichnet und versteht ganze Sätze. Hervorzuheben sind auch die einzeln handelnden Personen im Spiel: Thorin, Gandalf, Gollum und die Orks laufen im Spiel selbständig umher. Der Spieler kann mit diesen Personen sogar Unterhaltungen führen. So etwa in der Form: »SAG THORIN ⟩GIB MIR DEN SCHLÜSSEL⟨«. Allerdings ist hierbei noch die Frage gestellt, ob die Personen dieses auch immer tun. Man muß manchmal schon hartnäckig darauf bestehen, um etwas zu erreichen... »SAG THORIN ⟩HOLE BIER⟨, SAG THORIN ⟩HOLE BIER⟨, SAG THORIN ⟩HOLE BIER⟨«... Mit etwas Geduld wird man erreichen, was gewünscht ist.

»Den kleinen Hobbit« habe ich bereits vor einem Jahr parallel zu »Gordon Saga 1« programmiert. Daher ähnelt sich auch die Grafik der beiden Spiele stark. Zusammengesetzt aus den Standard-Zeichensätzen, aber oft umwerfend gut gelungen.

Für Tolkien-Fans gibt es hier einfach keinen Kompromiß: Abtippen und Spielen! (Michael Nickles/dm)

**Listing: »Der kleine Hobbit«**

```
0 REM ***********************************     <131>
1 REM *T H E  H O B B I T  MBM 19.8.84*       <196>
2 REM ***********************************     <133>
5 GOSUB 62500                                 <217>
11 I1=40960:I2=49151:RESTORE                  <244>
12 FOR I=I1 TO I2:POKE I,PEEK(I):NEXT:A=10
   *4096+8*256+10*16:FOR I=A TO A+2           <093>
13 READ X:POKE I,X:NEXT:A=704:FOR I=A TO A
   +5:READ X:POKE I,X:NEXT                    <089>
14 DATA 32,192,2                              <151>
15 DATA 32,138,173,76,247,183                 <152>
16 A=2*256+12*16+6:FOR I=A TO A+20:READ X:
   POKE I,X:NEXT                              <198>
17 DATA 208,3,76,29,168,32,192,2,32,19,166 <213>
18 DATA 56,165,95,233,1,164,96,76,36,168      <089>
19 POKE 40996,197:POKE 40997,2:POKE 1,54      <191>
23 PRINT"{CLR,WHITE}";:POKE 53280,0:POKE 5
   3281,0                                     <004>
24 DATA 4900,5000,5100,5300,5400,5700,5900
   ,5800,5500,5600,5300                       <135>
25 RESTORE 24:DIM OK(11):FOR I=1 TO 11:REA
   D II:OK(I)=II:NEXT I:OK=1                  <097>
26 DATA 7300,7400,7200,7100                   <094>
27 RESTORE 26:FOR I=1 TO 4:READ II:BU(I)=I
   I:NEXT:BU=1                                <099>
28 SM(1)=8600:SM(2)=8400:SM(3)=8200:SM(4)=
   8500:MS=0:SM=1:DIM L$(38)                  <016>
29 DATA DIE KARTE,2,DEN SCHLUESSEL,10,DAS
   SCHWERT,3400,DAS SEIL,3400                 <030>
30 DATA DEN RING,5900,DEN GOLDSCHLUESSEL,4
   700,DAS ESSEN,0,DEN BOGEN,6                <019>
31 DATA DAS FASS,7300,DEN GOLDSCHATZ,8600     <144>
32 DATA DER WARG,6100,THORIN,3000,GANDALF,
   3000,ELROND,3600,EINE WALDELBE,7000        <084>
33 DATA BARD,7700,GOLLUM,6000,DER BUTLER,7
   300,DER ORK,0,DER TROLL,3200               <168>
39 RESTORE 29:FOR I=1 TO 10:READ I$:READ I
   I:GE$(I)=I$:GE(I)=II:NEXT                  <072>
40 FOR I=1 TO 10:READ I$:READ II:PE$(I)=I$
   :PE(I)=II:NEXT                             <233>
80 REM   GE(1)=1:GE(2)=1:GE(3)=1:GE(4)=1:G
   E(5)=1:GE(6)=1:GL=3                        <158>
90 FA(0)=1:GL=0:BT=1:SA=1:ZM=3000:PE(2)=1:
   GOTO 2105                                  <179>
100 TI$="000000":L$="":POKE 198,0:PRINT"{D
    OWN,WHITE}> @";                           <078>
```

```
120 GET X$:IF PEEK(203)=1 THEN 160                <213>
121 IF TI$>"000030"THEN 160                        <160>
128 IF X$=""THEN 120                               <243>
129 IF L$=""AND ASC(X$)=20 THEN 120                <227>
130 I=ASC(X$):IF I<65 OR I>90 THEN IF I<>2
    0 AND I<>32 AND I<>34 THEN 120                 <033>
135 PRINT CHR$(20);X$;"@";                         <194>
137 IF ASC(X$)=20 THEN L$=LEFT$(L$,LEN(L$)
    -1):GOTO 120                                   <143>
140 L$=L$+X$:TI$="000000":IF LEN(L$)>76 TH
    EN FOR I=1 TO 80:PRINT CHR$(20);:NEXT:
    GOTO 115                                       <057>
150 GOTO 120                                       <110>
160 PRINT".";:FOR I=1 TO LEN(L$)+4:PRINT C
    HR$(20);:NEXT                                  <030>
170 REM SATZ IN L$(0)-L$(10) GLIEDERN              <236>
175 FOR I=0 TO 10:L$(I)="":NEXT                    <151>
180 LC=0:FOR I=1 TO LEN(L$)                        <164>
185 IF MID$(L$,I,1)=" "THEN LC=LC+1:GOTO 1
    90                                             <031>
188 L$(LC)=L$(LC)+MID$(L$,I,1)                     <134>
190 NEXT I                                         <018>
200 GOTO 1000                                      <140>
500 PRINT"{LIG.RED}";A$:POKE 198,0:WAIT 19
    8,1:FOR I=1 TO 40:PRINT CHR$(20);:NEXT
    :R$=""                                         <067>
510 PRINT"{WHITE,UP}";:RETURN                      <117>
1000 REM A U S W E R T U N G                       <161>
1005 R$="":CA=-1:OB=0:AC=0:VE=0                    <141>
1010 CA=CA+1:E$=L$(CA)                             <092>
1030 IF E$="N"THEN R$="NORDEN":ZM=N                <133>
1040 IF E$="S"THEN R$="SUEDEN":ZM=S                <082>
1050 IF E$="W"THEN R$="WESTEN":ZM=W                <141>
1060 IF E$="O"THEN R$="OSTEN":ZM=O                 <034>
1065 IF E$="RUNTER"THEN R$="{5LEFT}RUNTER"
     :ZM=R                                         <225>
1070 IF E$="RAUF"THEN R$="{5LEFT}RAUF":ZM=
     H                                             <233>
1075 IF E$="NW"THEN R$="NORDWESTEN":ZM=NW          <211>
1080 IF E$="NO"THEN R$="NORDOSTEN":ZM=NE           <076>
1085 IF E$="SW"THEN R$="SUEDWESTEN":ZM=SW          <014>
1090 IF E$="SO"THEN R$="SUEDOSTEN":ZM=SO           <105>
1100 IF E$<>"WARTE"AND E$<>"WA"THEN 1105           <032>
1101 IF ZN=3300 AND TU(1)<>1 THEN PRINT"EI
     N NEUER TAG BRICHT AN.":TU(1)=1:GOTO
     1600                                          <157>
1102 PRINT"SIE WARTEN.":GOTO 1600                  <028>
1105 IF E$="SCHAU"THEN ZM=ZN:GOTO 2105             <053>
1110 IF E$="OEFFNE"THEN GOSUB 11100                <251>
1115 IF E$="SCHLIESSE"THEN GOSUB 11200             <179>
1116 IF E$="TOETE"THEN GOSUB 11500                 <071>
1120 IF E$="NIMM"THEN VE=1:GOSUB 11300             <031>
1122 IF E$="VERLIER"THEN VE=2:GOSUB 11300          <048>
1124 IF E$="I"OR E$="INVENTUR"THEN GOSUB 1
     1400                                          <059>
1125 IF E$="ISS"THEN IF GE(7)=1 OR GE(7)=Z
     N THEN PRINT"SIE ESSEN.":GE(7)=0:GOTO
     1600                                          <192>
1130 IF E$="SAG"OR E$="SAGE"THEN GOSUB 200
     00:GOTO 1600                                  <078>
1131 IF VL=2 AND E$="ZEIT"THEN VL=0:GL=2:G
     OTO 1600                                      <108>
1132 IF E$<>"GRABE"OR ZN<>5200 THEN 1135           <040>
1133 PRINT"{DOWN}SIE GRABEN IM SAND UND EN
     TDECKEN EINE{3SPACE}FALLTUER.":TU(7)=
     1:GOTO 1600                                   <134>
1135 IF LEFT$(E$,5)="ZERST"OR E$="VERNICHT
     E"OR LEFT$(E$,5)="ZERSC"THEN AC=AC+1          <231>
1140 IF E$="SPINNENNETZ"OR E$="NETZ"THEN A
     C=AC+1                                        <209>
1142 IF E$="WIRF"THEN VE=17                        <069>
1143 IF E$="SEIL"AND GE(4)=1 THEN OB=4             <089>
1144 IF LEFT$(E$,4)="ZIEH"THEN VE=2                <070>
1146 IF LEFT$(E$,9)="UEBERQUER"OR LEFT$(E$
     ,3)="GEH"THEN VE=3                            <028>
1148 IF LEFT$(E$,7)="SCHWIMM"AND ZN=6900 T
     HEN VL=5                                      <039>
1150 IF E$="FASS"THEN IF GE(9)=1 OR GE(9)=
     ZN THEN OB=5                                  <127>
1152 IF LEFT$(E$,6)="SPRING"THEN VE=4              <147>
1500 IF CA=LC THEN 1600                            <177>
1510 GOTO 1010                                     <212>
1600 :                                             <052>
2000 REM A C T I O N                               <043>
2005 IF GE(10)=1 THEN 62000                        <148>
2010 IF ZM=0 AND R$<>""THEN A$="KEIN WEG I
     N DIESE RICHTUNG !":GOSUB 500:GOTO 22
     00                                            <181>
2020 IF ZM=0 THEN 2200                             <210>
2100 REM NEUES ZN                                  <226>
2103 PRINT"SIE GEHEN NACH ";R$;"."                 <001>
2105 ZN=ZM:ZM=0:RESTORE ZN:PRINT                   <167>
2110 READ DL                                       <082>
2111 FOR I=1 TO DL:READ D$:PRINT"{YELLOW}"
     ;D$;:NEXT:PRINT"{WHITE}":PRINT                <143>
2115 N=0:S=0:W=0:O=0:H=0:R=0:NW=0:NE=0:SW=
     0:SO=0:GOSUB ZN:GOSUB 10000:GOTO 2400         <201>
2200 REM                                           <230>
2204 IF VL>0 THEN 39000                            <078>
2400 REM DIE MONSTER BEWEGEN SICH                  <140>
2401 IF INT(10*RND(1))>5 THEN 2403                 <075>
2402 IF ZN=7000 THEN PRINT"{DOWN}DIE WALDE
     LBE NIMMT SIE GEFANGEN.":ZM=7100:GOTO
     2105                                          <225>
2403 :                                             <093>
2410 IF PE(9)=ZN THEN 2455                         <211>
2411 OK=OK+1:IF OK=12 THEN OK=1                    <037>
2415 PE(9)=OK(OK)                                  <016>
2420 IF PE(9)<>4900 THEN 2450                      <137>
2422 IF TU(5)<>0 THEN 2430                         <051>
2424 TU(5)=1:IF ZN=4100 THEN PRINT"{DOWN}I
     N DER HOEHLE OEFFNET SICH EIN SPALT !
     ":N=4900                                      <169>
2425 GOTO 2450                                     <055>
2430 TU(5)=0:IF ZN=4100 THEN PRINT"{DOWN}D
     ER SPALT SCHLIESST SICH.":N=0                 <066>
2450 IF PE(9)<>ZN THEN 2480                        <153>
2455 PRINT"{DOWN}EIN ORK BETRITT DEN RAUM
     !"                                            <016>
2460 I=INT(12*RND(1))+1                            <114>
2462 IF I<=2 OR GE(5)=1 THEN PRINT"{DOWN}D
     ER ORK GEHT WEITER.":GOTO 2411                <197>
2463 IF I>9 THEN 2480                              <209>
2464 IF I>2 THEN PRINT"{DOWN}DER ORK NIMMT
     SIE GEFANGEN.":ZM=5200:GOTO 2105              <207>
2480 IF BT=0 THEN 2495: REM BUTLER                 <004>
2481 IF ZN<>PE(8)THEN BU=BU+1:IF BU=5 THEN
     BU=1                                          <015>
2482 PE(8)=BU(BU)                                  <115>
2483 IF PE(8)<>7100 THEN 2490                      <092>
2484 IF TU(10)=1 THEN 2487                         <053>
2485 TU(10)=1:IF PE(8)=ZN THEN PRINT"{DOWN
     }JEMAND OEFFNET DIE KERKERTUER.":O=72
     00:BU=1                                       <212>
2486 GOTO 2490                                     <245>
2487 TU(10)=0:IF PE(8)=ZN THEN PRINT"{DOWN
     }JEMAND SCHLIESST DIE KERKERTUER.":O=
     0                                             <152>
2490 IF PE(8)<>ZN OR ZN=7100 THEN 2495             <034>
2491 PRINT"{DOWN}DER BUTLER DES ELBENKOENI
     GS ERSCHEINT{3SPACE}PLOETZLICH !"             <115>
2492 IF INT(10*RND(1))>4 THEN 2494                 <146>
2493 PRINT"{DOWN}DER BUTLER NIMMT SIE GEFA
     NGEN.":ZM=7100:GOTO 2105                      <254>
2494 PE(8)=BU(BU)                                  <127>
2495 IF SA=0 THEN 2500                             <191>
2496 MS=MS+1:IF MS=5 THEN MS=1                     <204>
2497 IF SM=ZN THEN GOSUB 34000:VL=6:DR=1:G
     OTO 2500                                      <037>
2498 SM=SM(MS):IF SM=ZN THEN DR=1                  <058>
2500 GOSUB 9000:REM T H O R I N                    <186>
2501 IF PE(3)=ZN THEN GOSUB 9500:REM GANDA
     LF                                            <002>
2502 IF PE(4)=ZN THEN GOSUB 30000:REM ELRO
     ND                                            <162>
2503 IF PE(7)=ZN THEN GOSUB 31000:REM GOLL
     UM                                            <058>
2504 IF PE(1)=ZN THEN GOTO 32000:REM WARG          <009>
2505 IF BA=1 THEN GOSUB 33000:REM BARD             <113>
2506 IF DR=1 AND SA=1 THEN GOSUB 34000:REM
     SMAUG                                         <092>
2509 IT=0:IF AC=2 AND ZN=6800 THEN IT=1:IF
     GE(3)=1 THEN TU(8)=1:O=6900:IT=2              <052>
2510 IF IT=1 AND GE(3)<>1 THEN A$="SIE HAB
     EN KEIN SCHWERT !":GOSUB 500:RETURN           <032>
2511 IF IT=2 THEN PRINT"{DOWN}SIE ZERSCHLA
     GEN DAS SPINNENNETZ MIT DEM SCHWERT."         <066>
2520 IF VE<>17 OR OB<>4 OR ZN<>6900 THEN 2
     550                                           <223>
2521 IF TU(9)=1 THEN A$="DAS HABEN SIE BER
     EITS.":GOSUB 500:RETURN                       <095>
2530 PRINT"{DOWN}SIE WERFEN DAS SEIL UEBER
     DEN FLUSS.":FOR I=1 TO 2000:NEXT:GE(
     4)=6900                                       <074>
```

**Listing: »Der kleine Hobbit« (Fortsetzung)**

```
2540 PRINT"{DOWN}DAS SEIL VERFAENGT SICH I
     M BOOT.":TU(9)=1                     <071>
2550 IF VE<>2 OR OB<>4 AND GE(4)<>6900 OR
     TU(9)<>1 THEN 2570                   <151>
2555 PRINT"{DOWN}SIE ZIEHEN DAS BOOT ZU SI
     CH HERUEBER.":TU(9)=2:GE(4)=1        <033>
2570 IF VE<>3 OR TU(9)<>2 THEN 2580       <178>
2575 PRINT"{DOWN}SIE UEBERQUEREN DEN FLUSS
      MIT DEM BOOT.":PE(2)=7600:ZM=7000   <234>
2576 TU(9)=3:GOTO 2105                    <047>
2580 IF ZN<>7300 THEN 2600                <038>
2582 IF VE<>17 OR OB<>5 OR FA(0)<>1 THEN 2
     585                                  <189>
2583 PRINT"{DOWN}SIE WERFEN DAS FASS DURCH
      DIE FALLTUER.":GE(9)=7500           <207>
2585 IF VE<>4 OR FA(0)<>1 THEN 2600       <205>
2587 PRINT"{DOWN}SIE SPRINGEN DURCH DIE FA
     LLTUER IN DAS{2SPACE}FASS,DAS IM FLUS
     S SCHWIMMT."                         <155>
2590 FOR I=1 TO 3000:NEXT                 <222>
2591 PRINT"{DOWN}SIE TREIBEN AUF DEM FLUSS
      IN OESTLICHE{2SPACE}RICHTUNG.":PE(2)
     =7600                                <204>
2592 FOR I=1 TO 3000:NEXT                 <222>
2594 PRINT"{DOWN}IN EINER STARKEN KURVE WE
     RDEN SIE AUS{3SPACE}DEM FASS HERAUS A
     NS UFER";                            <252>
2595 PRINT" GESCHLEUDERT."                <250>
2597 ZM=7600:GOTO 2105                    <131>
2600 :                                    <036>
2899 GOTO 100                             <033>
2910 REM     S P I E L K A R T E          <253>
3000 DATA 3,SIE BEFINDEN SICH IN EINER KOM
     PFORTABEL EINGERICHTETEN HOEHLE ,    <203>
3010 DATA DIE EINER LANGEN  HALLE AEHNELT. <020>
3015 DATA 0                               <108>
3016 IF TU(0)=1 THEN O=3100               <249>
3030 GOSUB 60000:RETURN                   <087>
3100 DATA 1,DAS EINSAME TAL:N=3200:VL=0:GO
     SUB 60050:RETURN                     <040>
3200 DATA 1,LAGERPLATZ DER TROLLE         <077>
3202 IF TU(1)=-1 THEN VL=1:GOTO 39000     <215>
3203 IF TU(1)=0 THEN TU(1)=-1:GOSUB 60090 <042>
3205 IF TU(1)=1 THEN PE$(10)="DIE VERSTEIN
     ERTEN TROLLE"                        <140>
3206 IF TU(1)=1 AND GE(2)=10 THEN GE(2)=32
     00                                   <113>
3207 IF TU(1)=1 THEN 3230                 <137>
3210 PRINT"DER BOESE TROLL SAGT 'KANN MAN
     DEN WOHL KOCHEN ?'"                  <172>
3220 PRINT"DER FETTE TROLL MEINT 'ER WIRD
     KAUM EI- NEN MUNDVOLL GEBEN.'{DOWN}  <157>
3230 S=3100:N=3300:SO=3500                <105>
3231 IF TU(1)=-1 THEN VL=1                <253>
3240 RETURN                               <250>
3300 DATA 3,DUNKLER PFAD MIT FUSSSPUREN DE
     R TROLLE. ,                          <199>
3302 DATA NOERDLICH LIEGT DIE TUER ZUR HOE
     HLE DER TROLLE.:GOSUB 60310          <004>
3303 IF TU(3)=1 THEN N=3400               <147>
3305 S=3200:VL=0:RETURN                   <025>
3400 DATA 1,IN DER HOEHLE DER TROLLE:S=330
     0:RETURN                             <222>
3500 DATA 1,AM KLEINEN FLUSS:VL=0:NW=3200:
     SO=3600:TU(4)=0:RETURN               <122>
3600 DATA 1,RIVENDELL:NW=3500:O=3700:RETUR
     N                                    <059>
3700 DATA 1,IM NEBELGEBIRGE:W=3600:N=3800:
     O=4300:RETURN                        <207>
3800 DATA 1,IM NEBELGEBIRGE:S=3700:N=3900:
     RETURN                               <247>
3900 DATA 1,IM NEBELGEBIRGE:S=3800:NE=4000
     :RETURN                              <196>
4000 DATA 1,IM NEBELGEBIRGE:SW=3900:SO=440
     0:RETURN                             <143>
4100 DATA 1,TROCKENE HOEHLE IM NEBELGEBIRG
     E:IF TU(5)=1 THEN N=4900             <153>
4110 S=4200:RETURN                        <020>
4200 DATA 1,IM NEBELGEBIRGE:NW=3900:N=4100
     :S=4300:RETURN                       <123>
4300 DATA 1,IM NEBELGEBIRGE:W=3700:N=4200:
     O=4800:NW=3800:RETURN                <016>
4400 DATA 1,IM NEBELGEBIRGE:NW=4000:R=4500
     :RETURN                              <103>
4500 DATA 1,IM NEBELGEBIRGE:R=4600:RETURN <151>
4600 DATA 1,IM NEBELGEBIRGE:W=4700:RETURN <203>
4700 DATA 1,IM NEBELGEBIRGE:SW=4300:O=4600
     :RETURN                              <057>
4800 DATA 1,EIN SCHWERER UND GEFAEHRLICHER
      GEBIRGS- PFAD IM NEBELGEBIRGE       <193>
4810 W=4300:O=6100:GOSUB 60175:RETURN     <114>
4900 DATA 1,HOEHLENSYSTEM DER ORKS:S=4100:
     O=5000:NE=5300:RETURN                <053>
5000 DATA 1,HOEHLENSYSTEM DER ORKS:W=4900:
     N=5300:O=5100:RETURN                 <140>
5100 DATA 1,HOEHLENSYSTEM DER ORKS:W=5000:
     NW=5300:N=5400:RETURN                <051>
5200 DATA 1,GEFAENGNIS DER ORKS:RETURN    <025>
5300 DATA 1,HOEHLENSYSTEM DER ORKS        <044>
5310 W=5200:SW=4900:S=5000:SO=5100:O=5400:
     H=5600:RETURN                        <102>
5400 DATA 1,HOEHLENSYSTEM DER ORKS:W=5300:
     S=5100:H=5700:RETURN                 <130>
5500 DATA 1,HOEHLENSYSTEM DER ORKS:N=5800:
     O=5600:RETURN                        <033>
5600 DATA 1,HOEHLENSYSTEM DER ORKS:W=5500:
     R=5300:RETURN                        <030>
5700 DATA 1,HOEHLENSYSTEM DER ORKS:NW=5900
     :N=6000:RETURN                       <180>
5800 DATA 1,HOEHLENSYSTEM DER ORKS:S=5500:
     O=5900:RETURN                        <050>
5900 DATA 1,HOEHLENSYSTEM DER ORKS:W=5800:
     SO=5700:RETURN                       <212>
6000 DATA 1,FEUCHTE GROTTE:S=5700:RETURN  <183>
6100 DATA 1,EINE WALDLICHTUNG:W=4800:RETUR
     N                                    <143>
6300 DATA 1,WEITES GRASLAND AM FUSS DES GE
     BIRGES:N=6400:RETURN                 <235>
6400 DATA 1,BEORNS HAUS:GE(7)=6400:S=6300:
     O=6600:NE=6500:GOSUB 60345:RETURN    <102>
6500 DATA 1,DAS TOR ZU MIRKWALD:SW=6400:S=
     6600:O=6700:GOSUB 60140:RETURN       <214>
6600 DATA 1,AUF DEM NACHTWALDWEG          <058>
6610 W=6400:O=6800:S=6700:RETURN          <047>
6700 DATA 1,AUF DEM NACHTWALDWEG          <160>
6710 PRINT"{DOWN}SIE SEHEN RIESIEGE AUGEN
     AUF SIE HERUN- TERSTARREN.":VL=4:RETU
     RN                                   <177>
6800 DATA 1,AUF DEM NACHTWALDWEG          <004>
6810 W=6600:S=6700                        <237>
6820 IF TU(8)=1 THEN O=6900:RETURN        <005>
6830 PRINT"{DOWN}EIN RIESIGES SPINNENNETZ
     VERSPERRT DEN{2SPACE}WEG NACH OSTEN.{
     DOWN}":RETURN                        <133>
6900 DATA 1,DER SCHWARZE SCHNELLE FLUSS   <205>
6910 W=6800                               <178>
6920 IF TU(9)<2 THEN PRINT"{DOWN}AUF DER A
     NDEREN SEITE DES FLUSSES LIEGT EIN BO
     OT.{DOWN}"                           <157>
6930 RETURN                               <130>
7000 DATA 1,AUF DEM NACHTWALDWEG          <206>
7010 RETURN                               <210>
7100 DATA 1,IM KERKER DER WALDELBEN       <092>
7110 RETURN                               <054>
7200 DATA 1,IN EINEM GANG                 <143>
7210 W=7100:S=7400:RETURN                 <116>
7300 DATA 1,IM WEINKELLER DES ELBENKOENIGS <162>
7310 O=7400:GOSUB 60270:RETURN            <035>
7400 DATA 1,AN DER MAGISCHEN TUER         <206>
7410 N=7200:W=7300:RETURN                 <048>
7600 DATA 1,AM GROSSEN SEE                <016>
7610 O=7700:RETURN                        <154>
7700 DATA 1,DIE STADT ESGAROTH            <243>
7710 W=7600:N=7800:GOSUB 60220:RETURN     <252>
7800 DATA 1,AM SCHNELLEN FLUSS            <046>
7810 N=8000:S=7700:O=7900:RETURN          <117>
7900 DATA 1,AM WASSERFALL:GOSUB 60405     <046>
7910 W=7800:RETURN                        <009>
8000 DATA 1,WUESTENLAND                   <219>
8010 S=7800:H=8100:RETURN                 <045>
8100 DATA 1,DIE RUINEN VON DAL:GOSUB 60470 <059>
8110 R=8000:H=8200:RETURN                 <172>
8200 DATA 1,DER EINSAME BERG:GOSUB 63100  <209>
8210 N=8400:R=8100:NW=8300:NE=8500:RETURN <141>
8300 DATA 1,DER RABENHUEGEL               <140>
8310 SO=8200:RETURN                       <050>
8400 DATA 1,EINGANG ZUR DRACHENHOEHLE     <189>
8410 S=8200:RETURN                        <235>
8500 DATA 1,TROCKENE HOEHLE               <187>
8505 IF FA(1)=1 THEN R=8600               <031>
8510 SW=8200:RETURN                       <059>
8600 DATA 1,DIE SCHLAFKAMMER DES DRACHEN  <009>
```

```
8610 SW=8400:RETURN                                   <032>
9000 IF PE(2)=5200 AND ZN<>5200 AND ZN<610
     0 THEN 9050                                      <075>
9001 REM  T H O R I N                                 <066>
9002 IF ZN>4900 AND ZN<6300 THEN 9050                 <127>
9003 IC=0:IF PE(2)<>ZN THEN R$="{DOWN}THOR
     IN KOMMT HERBEI.":PE(2)=ZN:IC=1                  <170>
9004 IF IC=1 AND PE(3)<>ZN AND PE(3)<>0 TH
     EN PE(3)=ZN:PE(2)=ZN:R$="BEIDE"                  <201>
9005 IF R$=""THEN 9009                                <081>
9006 IF R$="BEIDE"THEN R$="{DOWN}THORIN UN
     D GANDALF KOMMEN HERBEI."                        <034>
9007 PRINT R$                                         <175>
9009 IF GE(1)=2 AND ZN=3000 THEN PRINT"{DO
     WN}THORIN GIBT IHNEN DIE KARTE.":GE(1
     )=1                                              <116>
9010 IF INT(10*RND(1))>4 THEN RETURN                  <171>
9011 IF PE(3)=3100 AND ZN=3000 THEN PRINT"
     {DOWN}THORIN SAGT 'BEEIL DICH !'":GOT
     O 9016                                           <211>
9015 IF INT(10*RND(1))=5 THEN PRINT"{DOWN}
     THORIN SAGT 'BEEIL DICH !'":GOTO 9020 <028>
9016 IF INT(20*RND(1))=5 THEN PRINT"{DOWN}
     THORIN BEGINNT VON GOLD ZU SINGEN.":G
     OTO 9020                                         <154>
9017 IF INT(20*RND(1))=5 THEN PRINT"{DOWN}
     THORIN SETZT SICH UND WARTET."                   <097>
9018 IF INT(20*RND(1))<>5 OR PE(3)<>0 THEN
      9020                                            <122>
9019 PRINT"{DOWN}THORIN SAGT'WAERE GANDALF
      NUR BEI UNS.'.                                  <190>
9020 IF ZN>=4900 AND ZN<6000 AND INT(20*RN
     D(1))+1>3 AND PE(2)<>5200 THEN PE(2)=
     4800                                             <213>
9050 IF PE(3)<3700 THEN 9055                          <123>
9051 PRINT"{DOWN}GANDALF SAGT 'WICHTIGE AU
     FGABEN WARTEN{2SPACE}AUF MICH{SHIFT-S
     PACE}!' UND GEHT FORT."                          <189>
9052 PE(3)=0                                          <200>
9055 IF ZN=5200 AND PE(2)<>5200 THEN PRINT
     "{DOWN}THORIN KOMMT HEREIN.":PE(2)=52
     00                                               <199>
9060 IF GE(6)=1 THEN PRINT"{DOWN}THORIN NI
     MMT DEN GOLDSCHLUESSEL.":GE(6)=2                 <091>
9070 IF SA<>0 OR ZN<>8500 OR FA(1)<>0 THEN
      9499                                            <250>
9075 PRINT"{DOWN}EIN ROTER SONNENSTRAHL TR
     IFFT AUF DIE{3SPACE}GLATTE FELSWAND I
     N DER";                                          <236>
9076 PRINT" HOEHLE !":FOR I=1 TO 2000:NEXT <151>
9080 PRINT"{DOWN}EIN FLACHES STUECK GESTEI
     N SPLITTERT VOMFELS UND FAELLT HERUNT
     ER."                                             <087>
9090 PRINT"{DOWN}EIN KLEINES LOCH ERSCHEIN
     T ETWA DREI{4SPACE}FUSS UEBER DER ERD
     E !"                                             <102>
9095 PRINT"{DOWN}THORIN STECKT DEN GOLDSCH
     LUESSEL IN DAS LOCH."                            <223>
9097 PRINT"{DOWN}EINE TUER TUT SICH AUF -
     EIN STEILER WEGFUEHRT VON IHR HINAB I
     N ";                                             <229>
9098 PRINT"DIE DUNKELHEIT.":FA(1)=1:R=8600 <191>
9499 RETURN                                           <159>
9500 REM  G A N D A L F                               <068>
9503 IF UN(0)=3 THEN UN(0)=0:PRINT"{DOWN}G
     ANDALF SAGT 'WAS SOLL DER QUATSCH ?'" <042>
9504 IF INT(10*RND(1))>4 THEN RETURN                  <188>
9505 IF ZN=3000 AND TU(0)=1 THEN PRINT"{DO
     WN}GANDALF GEHT NACH OSTEN.":PE(3)=31
     00                                               <231>
9510 IF ZN=3000 AND TU(0)=0 THEN PRINT"{DO
     WN}GANDALF OEFFNET DIE TUER.":TU(0)=1
     :O=3100                                          <006>
9512 IF ZN<>3100 OR GA(0)=1 THEN 9514                 <147>
9513 PRINT"{DOWN}GANDALF SAGT 'EIN GROSSES
      ABENTEUER STE-HT DIR BEVOR !'":GA(0)
     =1                                               <232>
9514 :                                                <092>
9517 IF INT(20*RND(1))=1 AND GE(1)=1 THEN
     GE(1)=3:PRINT"{DOWN}GANDALF NIMMT DIE
      KARTE."                                         <166>
9519 IF INT(20*RND(1))=1 AND GE(1)=3 THEN
     GE(1)=1:PRINT"{DOWN}GANDALF GIBT DIR
     DIE KARTE.                                       <092>
9520 IF INT(30*RND(1))=1 THEN PRINT"{DOWN}
     GANDALF SAGT 'WAS NUN ?'."                       <223>
9999 RETURN                                           <151>
```

```
10000 REM MOEGLICHE RICHTUNGEN
                                                      <171>
10010 R$="":IF N THEN R$=R$+"N,"                      <037>
10011 IF S THEN R$=R$+"S,"                            <238>
10012 IF O THEN R$=R$+"O,"                            <142>
10013 IF W THEN R$=R$+"W,"                            <080>
10014 IF NW THEN R$=R$+"NW,"                          <078>
10015 IF NE THEN R$=R$+"NO,"                          <044>
10016 IF SW THEN R$=R$+"SW,"                          <251>
10017 IF SO THEN R$=R$+"SO,"                          <151>
10018 IF H THEN R$=R$+"RAUF,"                         <144>
10019 IF R THEN R$=R$+"RUNTER,"                       <226>
10020 IF R$=""THEN 10100                              <190>
10030 PRINT"MOEGLICHE RICHTUNGEN: ";LEFT$(
      R$,LEN(R$)-1):PRINT:R$=""                       <208>
10100 REM SIE SEHEN ...                               <111>
10105 IC=0:S$="{12RIGHT}"                             <047>
10110 PRINT"SIE SEHEN :{SPACE,UP}"                    <084>
10120 FOR I=1 TO 10:IF PE(I)<>ZN THEN 1012
      5                                               <156>
10121 IC=1:I$=PE$(I):IF LEFT$(I$,3)="DER"T
      HEN I$=MID$(I$,4,LEN(I$)):I$="EINEN"
      +I$                                             <179>
10123 PRINT S$;I$                                     <230>
10125 IF GE(I)=ZN THEN PRINT S$;GE$(I):IC=
      1                                               <000>
10130 NEXT                                            <234>
10190 IF ZN=3000 THEN IC=1:PRINT S$"DIE TR
      UHE                                             <180>
10200 IF ZN=3000 THEN IC=1:PRINT S$"DIE TU
      ER                                              <159>
10201 IF ZN=6300 THEN IC=1:PRINT S$"DEN CA
      RROCK                                           <055>
10202 IF ZN=3300 THEN IC=1:PRINT S$"DIE TU
      ER ZUR TROLLHOEHLE                              <068>
10210 IF ZN=4100 AND TU(5)=1 THEN IC=1:PRI
      NT S$"EINEN SPALT IN DER NORDWAND."  <197>
10220 IF ZN=5200 THEN IC=1:PRINT S$"EIN FE
      NSTER IM WESTEN.                                <054>
10222 IF ZN=5200 THEN IC=1:PRINT S$"DIE GE
      FAENGNISTUER.                                   <239>
10223 IF ZN=5200 THEN IC=1:PRINT S$"DEN SA
      NDBODEN                                         <068>
10224 IF TU(7)=1 AND ZN=5200 THEN PRINT S$
      "DIE FALLTUER"                                  <221>
10225 IF ZN=7300 THEN PRINT S$"DIE FALLTUE
      R"                                              <175>
10226 IF SA=0 AND ZN=MS THEN PRINT S$"DEN
      TOTEN SMAUG."                                   <037>
10227 IF ZN=7300 THEN PRINT"{DOWN}UNTER DE
      R FALLTUER RAUSCHT EIN STARKER{2SPAC
      E}FLUSS."                                       <188>
10228 IF ZN=8500 AND FA(1)=1 THEN PRINT S$
      "EINE OEFFENE TUER"                             <186>
10229 IF TU(9)=2 AND ZN=6900 THEN PRINT S$
      "EIN BOOT"                                      <163>
10230 IF ZN=7100 THEN PRINT S$"TUER NACH O
      STEN"                                           <187>
10290 IF IC=0 THEN PRINT"{11SPACE,11LEFT,U
      P}";                                            <205>
10300 FOR I=1 TO 10                                   <030>
10310 IF GE(I)=0 OR GE(I)>=3000 THEN 10350 <098>
10320 :FOR II=2 TO 10                                 <220>
10325 ::IF GE(I)=II AND PE(II)=ZN THEN PRI
      NT"{DOWN}";PE$(II);" HAT ";GE$(I);".
      "                                               <118>
10330 :NEXT II                                        <099>
10350 NEXT I                                          <018>
10500 REM IFIC=1THENPRINT                             <025>
10600 RETURN                                          <244>
11000 REM ROUTINEN FUER VERBEN                        <114>
11010 OB=0:FOR I1=0 TO LC:E$=L$(I1)                   <239>
11020 IF E$="TUER"THEN OB=1                           <186>
11030 IF E$="TRUHE"THEN OB=2                          <117>
11031 IF E$="FENSTER"THEN OB=3                        <007>
11032 IF E$="FALLTUER"THEN OB=4                       <125>
11040 NEXT I1:RETURN                                  <082>
11100 REM OEFFNE ********                             <128>
11110 GOSUB 11000                                     <184>
11114 IF OB<>2 THEN 11125                             <166>
11115 IF OB=2 AND ZN<>3000 THEN A$="HIER I
      ST KEINE TRUHE.":GOSUB 500:RETURN               <072>
11117 IF TU(2)=0 THEN PRINT"SIE OEFFNEN DI
      E TRUHE.{DOWN}.":TU(2)=1:RETURN                 <179>
11118 IF TU(2)=1 THEN A$="DIE TRUHE IST BE
      REITS OFFEN.":GOSUB 500:RETURN                  <024>
11125 IF OB<>1 OR ZN<>3000 THEN 11128                 <035>
```

**Listing: »Der kleine Hobbit« (Fortsetzung)**

```
11126 IF TU(0)=1 THEN A$="DIE TUER IST BER
      EITS OFFEN.":GOSUB 500:RETURN       <205>
11127 IF TU(0)=0 THEN PRINT"SIE OEFFNEN DI
      E TUER.":TU(0)=1:O=3100:RETURN      <239>
11128 IF ZN<>3300 THEN 11140              <140>
11130 IF TU(3)=0 AND GE(2)=1 THEN PRINT"SI
      E OEFFNEN DIE TUER.":TU(3)=1:N=3400:
      RETURN                              <149>
11131 IF TU(3)=0 AND GE(2)<>1 THEN A$="SIE
       HABEN KEINEN SCHLUESSEL.":GOSUB 500
      :RETURN                             <163>
11132 IF TU(3)=1 THEN A$="DIE TUER IST BER
      EITS OFFEN.":GOSUB 500:RETURN       <147>
11140 IF OB=3 AND ZN=5200 THEN A$="SIE SIN
      D ZU KLEIN DAZU.":GOSUB 500:RETURN  <137>
11145 IF ZN<>7300 OR OB<>4 THEN 11160     <205>
11147 IF FA(0)=0 THEN FA(0)=1:PRINT"SIE OE
      FFNEN DIE FALLTUER.":RETURN         <106>
11148 IF FA(0)=1 THEN A$="DIE FALLTUER IST
       BEREITS OFFEN.":GOSUB 500:RETURN   <087>
11160 A$="SIE KOENNEN DAS NICHT OEFFNEN.":
      GOSUB 500:RETURN                    <204>
11200 REM SCHLIESSE  *********            <227>
11210 GOSUB 11000                         <028>
11214 IF OB<>2 THEN 11225                 <018>
11215 IF OB=2 AND ZN<>3000 THEN A$="HIER I
      ST KEINE TRUHE.":GOSUB 500:RETURN   <172>
11217 IF TU(2)=1 THEN PRINT"SIE SCHLIESSEN
       DIE TRUHE.":TU(2)=0:RETURN         <080>
11218 IF TU(2)=0 THEN A$="DIE TRUHE IST BE
      REITS ZU.":GOSUB 500:RETURN         <013>
11225 IF OB<>1 OR ZN<>3000 THEN 11230     <181>
11226 IF TU(0)=0 THEN A$="DIE TUER IST BER
      EITS ZU.":GOSUB 500:RETURN          <123>
11227 IF TU(0)=1 THEN PRINT"SIE SCHLIESSEN
       DIE TUER.":TU(0)=0:UN(0)=UN(0)+1:O=
      0:RETURN                            <118>
11230 IF ZN=3300 AND TU(3)=1 THEN PRINT"SI
      E SCHLIESSEN DIE TUER.":TU(3)=0:N=0:
      RETURN                              <074>
11231 IF ZN=3300 AND TU(3)=0 THEN A$="DIE
      TUER IST BEREITS ZU.":GOSUB 500:RETU
      RN                                  <255>
11232 IF ZN<>7300 THEN RETURN             <015>
11233 IF OB=4 AND FA(0)=1 THEN FA(0)=0:PRI
      NT"SIE SCHLIESSEN DIE FALLTUER.":RET
      URN                                 <148>
11235 IF OB=4 AND FA(0)=0 THEN A$="DIE FAL
      LTUER IST BEREITS ZU.":GOSUB 500:RET
      URN                                 <171>
11260 A$="DAS GEHT NICHT.":GOSUB 500:RETUR
      N                                   <033>
11300 REM NIMM / VERLIERE **********      <203>
11310 IM=0:S$="":G1=0:FOR I=0 TO LC:E$=L$(
      LC)                                 <119>
11320 FOR I1=1 TO 10                      <120>
11330 IF E$=MID$(GE$(I1),5,LEN(GE$(I1))-4)
       THEN IM=I1:I1=11:I=LC+1            <165>
11340 NEXT I1:NEXT I                      <150>
11350 S$=GE$(IM):G1=GE(IM)                <174>
11360 IF VE=1 AND G1=1 THEN A$="SIE HABEN
      DAS BEREITS.":GOSUB 500:RETURN      <210>
11365 IF VE=1 AND G1=ZN THEN PRINT"SIE NEH
      MEN ";S$;".":GE(IM)=1:RETURN        <251>
11366 IF VE=1 THEN A$="SIE KOENNEN DAS NIC
      HT NEHMEN.":GOSUB 500:RETURN        <211>
11370 IF VE=2 AND G1=1 THEN GE(IM)=ZN:PRIN
      T"SIE VERLIEREN ";S$;".":RETURN     <184>
11375 A$="SIE HABEN DAS NICHT.":GOSUB 500:
      RETURN                              <253>
11400 REM INVENTUR ************           <148>
11410 PRINT"SIE HABEN :{UP}"              <221>
11420 IC=0:FOR I=1 TO 10:IF GE(I)=1 THEN P
      RINT,"{2RIGHT}"GE$(I):IC=1          <104>
11430 NEXT I                              <082>
11440 IF IC=0 THEN PRINT,"{2RIGHT}NICHTS" <037>
11450 RETURN                              <076>
11500 REM TOETE *********                 <158>
11505 P$="":P=0:FOR I=1 TO LC:E$=L$(LC)   <159>
11506 IF E$="THORIN"AND PE(2)=ZN THEN P$="
      THORIN"                             <076>
11507 IF E$="GANDALF"AND PE(3)=ZN THEN P$=
      "GANDALF"                           <057>
11508 IF E$="ELROND"AND PE(4)=ZN THEN P$="
      ELROND"                             <019>
11509 IF E$="WALDELBE"AND PE(5)=ZN THEN P$
      ="DIE WALDELBE"                     <014>
11510 IF E$="GOLLUM"AND PE(7)=ZN THEN P$="
      GOLLUM"                             <108>
11511 IF E$="ORK"AND PE(9)=ZN THEN P$="ORK
      "                                   <130>
11512 IF E$="TROLL"AND PE(10)=ZN THEN P$="
      TROLL"                              <060>
11513 IF E$="BUTLER"AND PE(8)=ZN THEN P$="
      DER BUTLER"                         <045>
11519 NEXT I                              <171>
11520 IF P$=""THEN A$="ICH SEH DAS HIER NI
      CHT":GOSUB 500:RETURN               <130>
11540 PRINT"{DOWN}SIE GREIFEN "P$" AN !"  <008>
11550 IF P$="DER BUTLER"THEN 11570        <081>
11552 PRINT:PRINT P$" VERTEIDIGT SICH ZU S
      EHR !"                              <020>
11554 FOR I=1 TO 2000:NEXT                <168>
11556 PRINT:PRINT P$" HAT SIE GETOETET.":V
      L=1.4:RETURN                        <053>
11570 IF GE(3)<>1 THEN PRINT"{DOWN}DER BUT
      LER HAT SIE GETOETET.":VL=1.4:RETURN <174>
11572 PRINT"{DOWN}SIE HABEN DEN BUTLER MIT
       EINEM SCHLAG{3SPACE}ENTHAUPTET.":BT
      =0                                  <212>
11574 PE$(8)="DEN TOTEN BUTLER":RETURN    <044>
20000 REM    S P R E C H E N              <222>
20005 IC=0:FOR I=2 TO 7                   <141>
20010 IF PE$(I)=L$(1)THEN IC=1            <195>
20011 IF PE(I)=ZN AND IC=1 THEN P$=PE$(I):
      P=I:IC=2                            <067>
20015 NEXT I                              <031>
20020 IF IC=1 THEN A$=""+L$(1)+" IST NICHT
       ANWESEND.":GOSUB 500:RETURN        <099>
20022 IF IC=0 THEN A$="DAS GEHT NICHT.":GO
      SUB 500:RETURN                      <160>
20030 PRINT"SIE SAGEN ETWAS ZU ";P$;"."   <040>
20040 I1=0:I2=0:FOR I=1 TO LEN(L$)        <054>
20042 IF MID$(L$,I,1)<>CHR$(34)THEN 20045 <184>
20043 IF I1=0 THEN I1=I+1:GOTO 20045      <206>
20044 IF I2=0 THEN I2=I                   <121>
20045 NEXT I                              <061>
20050 L$=MID$(L$,I1,I2-I1)                <185>
20060 REM SATZ IN L$(0)-L$(10) GLIEDERN   <058>
20062 FOR I=0 TO 10:L$(I)="":NEXT         <226>
20064 LC=0:FOR I=1 TO LEN(L$)             <236>
20066 IF MID$(L$,I,1)=" "THEN LC=LC+1:GOTO
       20070                              <187>
20068 L$(LC)=L$(LC)+MID$(L$,I,1)          <202>
20070 NEXT I                              <086>
20100 REM ANALYSIEREN                     <220>
20105 EG=0:VE=0:GG=0:FOR I=0 TO LC:E$=L$(I
      )                                   <003>
20110 IF E$="LIES"OR E$="UNTERSUCHE"THEN V
      E=1                                 <233>
20120 IF E$="GIB"THEN VE=2                <179>
20121 IF E$="SMAUG"THEN EG=3              <214>
20130 IF LEFT$(E$,4)="TRAG"THEN VE=3      <099>
20140 IF LEFT$(E$,3)="GEH"THEN VE=4       <228>
20145 IF E$="NIMM"THEN VE=5               <056>
20150 IF E$="VERLIER"THEN VE=6            <173>
20160 IF E$="HILF"THEN VE=7               <040>
20170 IF E$="MIR"OR E$="MICH"THEN EG=1    <095>
20171 IF E$="OEFFNE"THEN VE=8             <024>
20172 IF E$="FENSTER"THEN EG=2            <250>
20173 IF E$="TOETE"THEN VE=8              <206>
20174 IF VL=2 AND E$="ZEIT"THEN VL=0:GL=2 <255>
20175 FOR I1=1 TO 10                      <083>
20176 IF E$=MID$(GE$(I1),5,LEN(GE$(I1))-4)
       THEN GG=1                          <017>
20180 NEXT I1                             <215>
20190 NEXT I                              <206>
20200 REM                                 <194>
20210 IF VE<>1 OR P<>4 OR GG<>1 THEN 20280 <204>
20212 IF GE(1)<>1 THEN 20500              <006>
20215 PRINT"{DOWN}ELROND BETRACHTET DIE KA
      RTE.":PRINT"{DOWN}ELROND SAGT ";    <119>
20220 IF IK=0 THEN IK=(INT(5*RND(1))+1)*10
      +20220                              <230>
20221 GOTO IK                             <189>
20230 PRINT"'NORDOESTLICH VON BEORNS{4SPAC
      E}HAUS LIEGT DAS TOR ZU MIRKWALD.'." <094>
20231 RETURN                              <223>
20240 PRINT"'NOERDLICH VOM CARROCK LIEGTDA
      S HAUS VON BEORN.'.":RETURN         <149>
20250 PRINT"'OESTLICH VOM GROSSEN SEE{3SPA
      CE}LIEGT DIE STADT ESGAROTH.'.":RETU
```

```
      RN                                           <006>
20260 PRINT"'OESTLICH VOM SCHNELLEN{5SPACE
      }FLUSS IST EIN WASSERFALL.'.":RETURN <161>
20270 PRINT"AUF DER NORDWESTSEITE DES EINS
      AMEN BERGES IST DER RABENHUEGEL.'."  <226>
20271 RETURN                               <007>
20280 IF VE=2 AND P=3 AND GE(GG)=3 THEN PR
      INT:PRINT PE$(3)" GIBT IHNEN ";GE$(G
      G)                                   <171>
20281 IF VE=2 AND P=3 AND GE(GG)=3 THEN GE
      (GG)=1                               <111>
20290 IF VE=3 AND EG=1 AND P=2 AND ZN=5200
       THEN PRINT"{DOWN}THORIN TRAEGT SIE.
      ":TR=1:RETURN                        <031>
20292 IF VE<>8 OR P<>2 OR EG<>2 OR TU(6)<>
      0 THEN 20295                         <194>
20293 PRINT"{DOWN}THORIN OEFFNET DAS FENST
      ER.":TU(6)=1:RETURN                  <002>
20295 IF P=2 AND VE=4 AND TR=1 THEN TR=0:Z
      M=5300:PRINT:R$="{6LEFT,SPACE}DURCH
      DAS FENSTER":RETURN                  <122>
20300 IF P=6 AND VE=7 THEN PRINT"{DOWN}BAR
      D SAGT 'ICH WERDE SMAUG TOETEN !'.":
      BA=1:RETURN                          <107>
20310 IF EG<>3 OR P<>6 OR VE<>8 OR SM<>ZN
      THEN 20320                           <229>
20311 PRINT"{DOWN}BARD NIMMT SEINEN SCHWAR
      ZEN PFEIL.":PRINT"{DOWN}EINE ALTE DR
      OSSEL ";                             <249>
20312 PRINT"KOMMT ANGEFLOGEN UND{2SPACE}SE
      TZT SICH AUF BARDS SCHULTER.         <153>
20313 PRINT"{DOWN}BARD SAGT 'SCHWARZER PFE
      IL,DU HAST NIE{2SPACE}GEFEHLT,GLUECK
       SEI MIT DIR.                        <117>
20314 FOR I=1 TO 2000:NEXT:PRINT"{DOWN}BAR
      D SCHIESST AUF SMAUG."               <172>
20315 FOR I=1 TO 3000:NEXT:IF INT(100*RND(
      1))<95 THEN 20317                    <196>
20316 PRINT"{DOWN}DER SCHWARZE PFEIL VERFE
      HLT SMAUG !":VL=6:RETURN             <059>
20317 PRINT"{DOWN}DER SCHWARZE PFEIL BOHRT
       SICH TIEF IN{3SPACE}SMAUGS BRUST !" <226>
20318 PRINT"{DOWN}SMAUG STUERZT UND STIRBT
      .":SA=0:MS=ZN:VL=0                   <063>
20320 RETURN                               <056>
20499 IF P=7 THEN 20600                    <083>
20500 PRINT"{DOWN}"P$" SAGT 'NEIN.'.       <195>
20600 RETURN                               <082>
30000 REM  E L R O N D                     <206>
30005 TU(4)=TU(4)+1                        <055>
30010 IF TU(4)=1 THEN PRINT"{DOWN}ELROND S
      AGT 'HALLO !'.                       <159>
30015 IF TU(4)=2 THEN PRINT"{DOWN}ELROND B
      IETET IHNEN ETWAS ZU ESSEN AN.":GE(7
      )=3600                               <056>
30020 RETURN                               <104>
31000 REM G O L L U M                      <009>
31005 IF GL>1 THEN 31100                   <109>
31010 PRINT"{DOWN}GOLLUM FRAGT 'ETWAS,DAS
      ALLES UND JEDEN VERSCHLINGT:BAUM DER
       RAU";                               <041>
31012 PRINT"SCHT,VOGEL DER{2SPACE}SINGT,FR
      ISST EISEN,ZERMALMT DEN HAERTES-TEN
      STEIN,";                             <231>
31014 PRINT"ZERBEISST JEDES SCHWERT,ZERB-
      RICHT JEDEN SCHREIN,SCHLAEGT ";      <073>
31015 PRINT"KOENIGE{4SPACE}NIEDER,SCHLEIFT
       IHREN PALAST,TRAEGT{5SPACE}MAECHTIG
      EN FELS";                            <204>
31017 PRINT" FORT ALS LEICHTE LAST?'.":VL=
      2:GL=2:RETURN                        <077>
31100 IF GL=2 THEN PRINT"{DOWN}GOLLUM SAGT
      'ER WEISS ES MEIN SCHATZZZ.'.":GL=3:
      RETURN                               <139>
31110 GL=GL+1:IF GL>5 AND GE(5)=1 THEN GE(
      5)=7:PRINT"{DOWN}GOLLUM NIMMT DEN RI
      NG.":RETURN                          <189>
31120 IF GE(5)=1 THEN PRINT"{DOWN}GOLLUM S
      AGT ' WAS HAT ER IN DER TASCHE{2SPAC
      E}MEIN SCHATZ?'.                     <255>
31130 IF GE(5)=7 THEN VL=2                 <068>
31140 RETURN                               <208>
32000 REM W A R G                          <033>
32010 PRINT"{DOWN}DER FUERCHTERLICHE WARG
      GREIFT SIE AN !"                     <062>
32020 IF GE(3)+GE(4)+GE(5)<>3 OR GL=0 THEN
       32100                               <119>
32030 PRINT"{DOWN}SIE VERTEIDIGEN SICH MIT
       GANZER KRAFT...                     <230>
32032 FOR I=1 TO 3000:NEXT                 <081>
32034 PRINT"EIN RIESIGER ADLER STUERZT AUS
       DER LUFT AUF SIE HERUNTER,PACKT SIE
       ";                                  <058>
32036 PRINT"MIT SEINEN{3SPACE}KLAUEN UND F
      LIEGT DAVON !                        <057>
32038 FOR I=1 TO 2000:NEXT:PE(2)=6300      <251>
32040 PRINT"{DOWN}NACH EINEM LANGEN FLUG L
      ANDET DER ADLER IN DER NAEHE EINES "
      ;                                    <087>
32042 PRINT"FELSPLATEAUS,LAESST{2SPACE}SIE
       FREI UND FLIEGT DAVON...":ZM=6300:G
      OTO 2105                             <188>
32100 VL=3:GOTO 39000                      <229>
33000 REM  B A R D                         <168>
33010 IF PE(6)<>ZN THEN PE(6)=ZN:PRINT"{DO
      WN}BARD FOLGT IHNEN."                <156>
33050 RETURN                               <087>
34000 REM  S M A U G                       <165>
34010 IF SM=ZN THEN PRINT"{DOWN}SMAUG ERSC
      HEINT !"                             <197>
34040 IF INT(20*RND(1))<5 THEN PRINT"{DOWN
      }SMAUG SAGT 'ICH KRIEG DICH,DIEB !'.
      "                                    <041>
34100 RETURN                               <121>
39000 REM V E R L O R E N                  <230>
39010 PRINT                                <249>
39015 IF VL=1 THEN PRINT"DER TROLL HAT SIE
       GEFRESSEN."                         <071>
39017 IF VL=2 THEN PRINT"JEMAND HAT SIE VO
      N HINTEN ERWUERGT.                   <080>
39019 IF VL=3 THEN PRINT"DER WARG HAT SIE
      GETOETET.                            <018>
39020 IF VL=4 THEN PRINT"ETWAS SCHWERES FA
      ELLT HERUNTER UND ER-{2SPACE}SCHLAEG
      T SIE."                              <022>
39021 IF VL=5 THEN PRINT"ALS SIE DAS WASSE
      R BERUHREN FALLEN SIE{2SPACE}IN SCHL
      AF....."                             <194>
39023 IF VL=6 THEN PRINT"SMAUG HAT SIE GEF
      RESSEN.                              <220>
39500 END                                  <131>
60000 REM G R A F I K
                                           <187>
60005 REM HOBBIT-HOEHLE                    <133>
60015 PRINT,"{BROWN,7SPACE}@@@@@@@",,,"{4S
      PACE}PP£{RVSON,ORANGE}M{2SPACE}B{2SP
      ACE}N{RVOFF,BROWN}LP",,," PP£{ORANGE
      ,RVSON}M{2SPACE}MM B NN N{RVOFF,BROW
      N}LP                                 <228>
60020 PRINT,"H{ORANGE}*{RVSON,2SPACE}MM{2S
      PACE}M{BROWN}RER{ORANGE}N NN{SPACE,R
      VOFF}£{BROWN}G",,,"N{SPACE,ORANGE}*{
      RVSON,2SPACE}M{BROWN}RERERERE{ORANGE
      }N{SPACE,RVOFF}£{SPACE,BROWN}H       <204>
60025 PRINT,"H{2SPACE,RVSON}RERERERERERER{
      RVOFF,2SPACE}G",,,"H{2SPACE,RVSON}ER
      ERERERERERE{RVOFF,2SPACE}G           <156>
60030 PRINT,"H{2SPACE,RVSON}RE{BLUE}OP{BRO
      WN}RE{RVOFF,ORANGE}*{RVSON}@@@{RVOFF
      }£{BROWN,RVSON}ER{RVOFF,2SPACE}G",,,
      "H{2SPACE,RVSON}ER{BLUE}HN{BROWN}ER{
      RIGHT,ORANGE}G{LEFT}GBB{RIGHT,BROWN}
      RE{RVOFF,2SPACE}G                    <175>
60035 PRINT,"H{2SPACE,RVSON}RE{GREEN}L@{BR
      OWN}RE{RIGHT,GREY 1}D{ORANGE}HH{RIGH
      T,BROWN}ER{RVOFF,2SPACE}G",,,"N{2SPA
      CE,RVSON}ERERER{RIGHT,ORANGE}GBB{RIG
      HT,BROWN}RE{RVOFF,2SPACE}G           <230>
60040 PRINT,"H{2SPACE,RVSON}RERERE{ORANGE}
      £TTT*{BROWN}ER{RVOFF,2SPACE}G",,,"N{
      SPACE,ORANGE,RVSON}£TTTTTT{5SPACE}TT
      *{RVOFF,BROWN,SPACE}G                <041>
60045 PRINT,"H{ORANGE,RVSON}£{15SPACE}*{RV
      OFF,BROWN}G":GOTO 63000              <214>
60050 REM WUESTENLAND                      <226>
60055 PRINT"{BROWN}","NYYYYYYYYYYYYYYYYYYY
      YH                                   <085>
60060 FOR I=1 TO 13:PRINT,"N",," H":NEXT   <156>
60065 PRINT"{14UP}","{DOWN,2RIGHT,DOWN,WHI
      TE}.{UP,2RIGHT}.{2RIGHT,2DOWN}.{2UP,
      RIGHT}.{2RIGHT}.{DOWN}.{2RIGHT}.{UP,
      RIGHT}.                              <157>
60070 PRINT,"{RIGHT,5DOWN,GREY 1}N{UP}N{UP
      }N{UP}NM{DOWN}MNMN{UP}NM{DOWN}MNMNM{
```

**Listing: »Der kleine Hobbit« (Fortsetzung)**

```
       2SPACE}NM",,"{6RIGHT}N N{3SPACE}N N{
       2SPACE}MN                              <199>
60075 PRINT,"{5RIGHT}N N{3SPACE}N N{3SPACE
       }N",,,,"{2SPACE,BROWN}MN",,,"{4RIGHT
       }WN{2SPACE}MN{2SPACE}WN",,,"{4RIGHT}
       MH{2SPACE}WN{2SPACE}H                  <136>
60080 PRINT,"{4RIGHT}W{3SPACE}W{2SPACE}W{O
       RANGE}*{RVSON,8SPACE,RVOFF}":FOR I=1
        TO 2:PRINT,"{RIGHT,RVSON,20SPACE,RV
       OFF}"                                  <166>
60085 NEXT:GOTO 63000                         <055>
60090 REM ORK-GRILLPLATZ                      <156>
60095 PRINT,"{BROWN,SPACE}GGGGGGGGGGGGGGGG
       GGG":FOR I=1 TO 14:PRINT,"W",,"G":NE
       XT                                     <088>
60100 PRINT," TTTTTTTTTTTTTTTTTTT"            <201>
60105 PRINT"{15UP}","{DOWN,2RIGHT,WHITE}.{
       3SPACE,DOWN}.{UP,2RIGHT}.{2RIGHT,2DO
       WN}.{2UP}.{DOWN,2RIGHT}.{2DOWN}.       <045>
60110 PRINT,,"{2UP,4LEFT,BROWN}MN MN{DOWN,
       5LEFT}NNMNMNMN",,,"{4RIGHT}MNMH MHN
       HNMMN                                  <239>
60115 PRINT,"HMN MNWNNWNNMHMHMNNG",,"NMHM
       NMNNMHMHMHNNNN MHH                     <177>
60120 PRINT,"HNNMHMH MHN MHNN NNG",,"{RIG
       HT}WN WNW{2SPACE}W{GREY 1}FFD{BROWN}
       W MHNMHH                               <169>
60125 PRINT,"HMGMMHNH{SPACE,GREY 1}V{BROWN
       }N{YELLOW}X{BROWN}M{GREY 1}F{BROWN,S
       PACE}WN WNG",,"WNN WN{3SPACE,GREY 1}
       E VV{BROWN,2SPACE}MH MHH               <231>
60130 PRINT,"HN{2SPACE}W{11SPACE}H{2SPACE}
       HG",,"W{6SPACE,ORANGE}TTTTTTTTT{BROW
       N,3SPACE}H                             <054>
60135 PRINT,"HGGGGGGGGGGGGGGGGGGGGG":GOTO 6
       3000                                   <040>
60140 REM DAS WALDTOR                         <216>
60145 PRINT,"{LIG.BLUE,RVSON,16SPACE,RVOFF
       }",,,"{RVSON,6SPACE}NMNM{6SPACE,RVOF
       F}                                     <010>
60150 PRINT,"{RVSON}M{2SPACE}NMN N{2SPACE}
       MNMNM{SPACE,RVOFF}",,,"{RVSON,SPACE}
       MN N N{RVOFF,YELLOW}NM{RVSON,LIG.BLU
       E,SPACE}N N{2SPACE}M{RVOFF}            <249>
60155 PRINT,"{RVSON,SPACE}N N N{RVOFF,YELL
       OW}NNMM{RVSON,LIG.BLUE,SPACE}N{4SPAC
       E,RVOFF}",,,"{RVSON}N{4SPACE,RVOFF,Y
       ELLOW}NN{2SPACE}MM{RVSON,LIG.BLUE,5S
       PACE,RVOFF}                            <170>
60160 PRINT,"{YELLOW}TTTTTOVYYYETTTTT",,,"
       {BROWN,RVSON}RERER{YELLOW,RVOFF}OOYY
       PP{BROWN,RVSON}RERER                   <071>
60165 PRINT,"{RVSON}ERERE{YELLOW,RVOFF}HH{
       RVSON,2SPACE,RVOFF}WN{BROWN,RVSON}ER
       ERE",,,"{RVSON}RERER{YELLOW,RVOFF}HH
       {RVSON,2SPACE,RVOFF}WN{BROWN,RVSON}R
       ERER                                   <008>
60170 PRINT,"{RVSON}ERERE{YELLOW,RVOFF}HH{
       RVSON,2SPACE,RVOFF}WN{BROWN,RVSON}ER
       ERE",,,"{RVSON,ORANGE,16SPACE,RVOFF}
       ":GOTO 63000                           <085>
60175 REM GEBIRGSPFAD                         <113>
60180 PRINT,"{BROWN,SPACE}GGGGGGGGGGGGGGGG
       GGG",,"W{4SPACE,WHITE}.{ORANGE,RVSON
       }HHH H H{2SPACE}H{2SPACE}B{SPACE,RVO
       FF,BROWN}G                             <177>
60185 PRINT,"H{SPACE,WHITE}.{ORANGE,3SPACE
       ,RVSON}HHH H H{2SPACE}H{2SPACE}B{SPA
       CE,RVOFF,BROWN}G",,"W{2SPACE,WHITE}.
       {ORANGE,2SPACE,RVSON}HHH H H{2SPACE}
       H{2SPACE}B{SPACE,RVOFF,BROWN}G         <245>
60190 PRINT,"H{WHITE}.{3SPACE}.{ORANGE,RVS
       ON}HHH H H{2SPACE}H{2SPACE}B{SPACE,R
       VOFF,BROWN}G",,,"W{SPACE,ORANGE,RVSON
       }£*{RVOFF,2SPACE,RVSON}HHH H H{2SPAC
       E}H{2SPACE}B{SPACE,RVOFF,BROWN}G       <144>
60195 PRINT,"H{ORANGE,RVSON}£NM*£{BROWN}YY
       P{ORANGE,RVOFF}*{RVSON}H H{2SPACE}H{
       2SPACE}B{SPACE,RVOFF,BROWN}G",,"W{OR
       ANGE,RVSON}N{2SPACE}MM{BROWN,2SPACE}
       W*{ORANGE,RVOFF,SPACE}*{RVSON}H{2SPA
       CE}H{2SPACE}B{SPACE,RVOFF,BROWN}H      <013>
60200 PRINT,"H{ORANGE,RVSON,SPACE}NM M{BRO
       WN,2SPACE}WN{SPACE,RVSON}*{RVOFF,SPA
       CE,ORANGE}*{RVSON,SPACE}H{2SPACE}B{S
       PACE,RVOFF,BROWN}G",,"W{ORANGE,RVSON
       }N{2SPACE}MN{BROWN,2SPACE}WN WN{RVSO
       N,CYAN,RVOFF,SPACE,ORANGE,SPACE}*{RV
       SON,2SPACE}B{SPACE,RVOFF,BROWN}G       <159>
60205 PRINT,"H{ORANGE,RVSON,3SPACE}N{SPACE
       ,BROWN,2SPACE}WN WN{SPACE,RVOFF,2SPA
       CE,ORANGE}*{RVSON,SPACE}B{SPACE,RVOF
       F,BROWN}G",,"W{GREY 1,RVSON,5SPACE,B
       ROWN,2SPACE}WN WN{SPACE,RVOFF,3SPACE
       ,ORANGE}*{RVSON}B{SPACE,RVOFF,BROWN}
       G                                      <007>
60210 PRINT,"H{GREY 1,RVSON,9SPACE,BROWN,S
       PACE}WN{SPACE,RVOFF,4SPACE,ORANGE}*{
       RVSON,SPACE,RVOFF,BROWN}G",,"H{GREY
       1,RVSON,13SPACE,RVOFF,5SPACE,ORANGE}
       *{BROWN}G                              <167>
60215 PRINT,"{SPACE,BROWN}TTTTTTTTTTTTTTTT
       TTT":GOTO 63000                        <146>
60220 REM FORRESTRIVER                        <038>
60225 FOR I=1 TO 6:PRINT,"{RVSON,LIG.BLUE,
       19SPACE,RVOFF}":NEXT                   <147>
60230 PRINT,"{RVSON,LIG.BLUE}GGGGGGGGGGGGG
       GGGGGG":FOR I=1 TO 7:PRINT,"{RVSON,L
       IG.GREEN,19SPACE}":NEXT                <193>
60235 PRINT,"{9UP,RVSON,LIG.BLUE}N{UP}N{UP
       }NM{DOWN}M{LEFT,3UP}UI{4DOWN,2LEFT}N
       {UP}N{UP}N{3UP}UI{DOWN}UI{3DOWN,2LEF
       T}N{UP}NM{DOWN,LEFT}N{UP}N{UP}NM{DOW
       N}MNMN                                 <158>
60240 PRINT,,"{3LEFT,2UP,LIG.BLUE}£*{DOWN,
       2LEFT,RVSON,GREY 1}UI{DOWN,2LEFT}JK{
       DOWN,2LEFT,2SPACE,RVOFF,3SPACE,LIG.B
       LUE}*{DOWN,6LEFT,RVSON,GREY 1,2SPACE
       ,GREY 2}HFFF{DOWN,6LEFT,GREY 1,2SPAC
       E,GREY 2}HVVV                          <218>
60245 PRINT,"{UP,6LEFT,LIG.GREEN}£*{RVSON
       ,GREY 1,DOWN,2LEFT,GREY 1}D{SPACE,RI
       GHT,GREY 1,RVOFF}F{RVSON}F{GREY 2}H{
       3SPACE,2RIGHT,UP,LIG.GREEN,RVOFF}£*{
       RVSON,DOWN,2LEFT,GREY 1}D ",,,"{RIGH
       T,RVOFF,LIG.GREEN}£*","{RIGHT}£*"      <126>
60250 PRINT,"{RIGHT,GREY 2,RVSON}D{SPACE,R
       VOFF,LIG.GREEN}£ *{RIGHT}£ *{RIGHT,R
       VSON,GREY 1,SPACE}F{RVOFF,RIGHT,LIG.
       GREEN}£ *",,,"{3RIGHT,GREY 1,RVSON}E
       D{SPACE,RIGHT,GREY 2,SPACE}D{SPACE,4
       RIGHT,SPACE}F "                        <014>
60255 PRINT,"{LIG.GREEN,RVSON,SPACE}EPOT{R
       VOFF}UYTY{RVSON}TOO{7SPACE,RVOFF}      <055>
60260 PRINT," EPOT{RVSON}UYTU{RVOFF}TP{ORA
       NGE,RVSON}£{RVOFF}£{LIG.GREEN}YU{RVS
       ON}TOP{SPACE,RVOFF}                    <143>
60265 PRINT,"{RVSON,12SPACE}TU{RVOFF}TOPG"    <119>
60266 PRINT,,"{11UP,5RIGHT,LIG.BLUE,RVSON}
       N{RIGHT}N{DOWN,4LEFT}N{3UP,3RIGHT,12
       DOWN}":GOTO 63000                      <227>
60270 REM WEINKELLER                          <079>
60275 PRINT,"{SPACE,GREY 2}GGGGGGGGGGGGGGG
       GG                                     <105>
60280 FOR I=1 TO 2:PRINT,"W{4SPACE,RVSON}E
       RERERERERERE{RVOFF}G":PRINT,"W{4SPAC
       E,RVSON}RERERERERERER{RVOFF}G":NEXT    <065>
60285 PRINT,"H{3SPACE,BROWN,RVSON}£{GREY 2
       }ERERE{ORANGE}HBG{GREY 2}E{ORANGE}HB
       G{GREY 2}E{RVOFF}G":PRINT,"W{2SPACE,
       BROWN,RVSON}£N{GREY 2}RERER{ORANGE}B
       BB{GREY 2}E{ORANGE}BBB{GREY 2}R{RVOF
       F}G"                                   <127>
60290 PRINT,"H{SPACE,BROWN,RVSON}£NN{RVOFF
       }GGGG{SPACE,ORANGE,RVSON}HBG{RIGHT}H
       BG{RVOFF,GREY 2,SPACE}G":PRINT,"H{SP
       ACE,BROWN,RVSON}NN{RVOFF,BROWN}£LPPN
       {9SPACE,GREY 2}G                       <113>
60295 PRINT,"H{SPACE,BROWN,RVSON}N{RVOFF,B
       ROWN}£{LIG.BLUE,RVSON}£{2SPACE,RVOFF
       }£{9SPACE,GREY 2,SPACE}G               <077>
60300 PRINT,"HN{BROWN}£{LIG.BLUE,RVSON}£{2
       SPACE,RVOFF}£{10SPACE,GREY 2,SPACE}G
       ":PRINT,"H{SPACE,BROWN}TTTT{5SPACE,O
       RANGE,RVSON}HBG{RVOFF,4SPACE,GREY 2}
       G                                      <217>
60305 PRINT,"H{6SPACE,ORANGE,RVSON}HBG{RIG
       HT}BBB{RVOFF,GREY 2,4SPACE}G":PRINT,
       " TTTTTTTTTTTTTTTTT":GOTO 63000        <244>
60310 REM DUNKLER PFAD                        <219>
60315 PRINT:FOR I=1 TO 3:PRINT,"{GREEN,RVS
       ON,14SPACE}":NEXT                      <034>
```

```
60320 PRINT,"C{BROWN}R{GREEN}UC{BROWN}R{SP
      ACE,GREEN}U{BROWN}R{GREEN}C{RVSON,2S
      PACE}DF "                            <086>
60325 FOR I=1 TO 3:PRINT,"{BROWN,SPACE}R{2
      SPACE}R{2SPACE}R{2SPACE}R{2SPACE}R":
      NEXT                                 <055>
60330 PRINT,"{ORANGE,RVSON,SPACE}@POIP POP
      @ @P"                                <146>
60335 PRINT,"@@POIP@POP@@@P"               <253>
60340 PRINT,"UUUUUUUUUUUUUU":GOTO 63000   <119>
60345 REM BEORNS HAUS                      <159>
60350 PRINT,"{LIG.BLUE,RVSON,16SPACE,RVOFF
      }                                    <158>
60355 PRINT,"{RVSON,3SPACE,SHIFT-SPACE,6SP
      ACE}NM{4SPACE,RVOFF}                 <070>
60360 PRINT,"{RVSON,SPACE}NM{2SPACE}NM{RVO
      FF}£*{RVSON}N{2SPACE}M{2SPACE}N{RVOF
      F}                                   <046>
60365 PRINT,"{RVSON}N{2SPACE}MN{SPACE,RVOF
      F}£{2SPACE,RVSON,ORANGE,SPACE,LIG.BL
      UE,3SPACE}MN{SPACE,RVOFF}            <104>
60370 PRINT,"{RVSON,3SPACE}N{SPACE,RVOFF}£
      {3SPACE,RVSON,ORANGE,SPACE,LIG.BLUE,
      RVOFF}*{RVSON,2SPACE}N{2SPACE,RVOFF} <024>
60375 PRINT,"{RVSON,2SPACE}N{SPACE,RVOFF}£
      {6SPACE}*{RVSON,4SPACE,RVOFF}        <025>
60380 PRINT,"{RVSON,5SPACE,BROWN}ERERER{LI
      G.BLUE,5SPACE,RVOFF}                 <169>
60385 PRINT,"{RVSON,5SPACE,BROWN}RE{RIGHT}
      E{WHITE,SPACE,BROWN}E{LIG.BLUE,5SPAC
      E,RVOFF}                             <248>
60390 PRINT,"{RVSON}RRRRR{BROWN}ER{RVOFF,O
      RANGE}V{RVSON,BROWN}RER{LIG.BLUE}RRR
      RR{RVOFF}                            <043>
60395 PRINT,"{RVSON,GREEN}TTTTTTTTTTTTTTTT
      {RVOFF}                              <181>
60400 PRINT,"{RVSON,16SPACE,RVOFF}":GOTO 6
      3000                                 <122>
60405 REM WASSERFALL                       <193>
60410 PRINT,"{SPACE,BROWN}@@@@@@@@@@@@@@@@ <210>
60415 PRINT,"R H H YRR{LIG.BLUE}G{BROWN}GG
      B B R                                <149>
60420 PRINT,"R Y B B R{LIG.BLUE,RVSON,SPAC
      E,RVOFF,BROWN,SPACE}RH B R           <077>
60425 PRINT,"R H B GN{LIG.BLUE}R{RVSON,SPA
      CE,RVOFF,BROWN}GLN H R               <142>
60430 PRINT,"R H G GH{LIG.BLUE}N{RVSON,SPA
      CE,RVOFF}R{BROWN}GN N R              <163>
60435 PRINT,"R B G GB{LIG.BLUE}T{RVSON,SPA
      CE,RVOFF}T{BROWN}B R RR              <241>
60440 PRINT,"R{GREEN}N{BROWN}B{GREEN}N{BRO
      WN}RN H{LIG.BLUE,RVSON}R{SPACE,RVOFF
      }R{BROWN}Y G GH                      <153>
60445 PRINT,"R{GREEN,RVSON}T{RVOFF}R{RVSON
      }RT{RVOFF,BROWN}Y N{LIG.BLUE,RVSON}T
      T{RVOFF,BROWN}H G RR                 <070>
60450 PRINT,"R{GREEN,RVSON}RTRN{RVOFF,BROW
      N}H H{LIG.BLUE,RVSON,3SPACE,RVOFF,BR
      OWN}N BN R                           <219>
60455 PRINT,"R{GREEN,RVSON,SPACE}R{2SPACE,
      RVOFF,BROWN}B{BLUE}G{RVSON}TTTTT{RVO
      FF}G{BROWN}BN R                      <240>
60460 PRINT,"RT T{BLUE}G{RVSON,9SPACE,RVOF
      F}G{BROWN}RR                         <044>
60465 PRINT," TTTTTTTTTTTTTTT":GOTO 63000  <107>
60470 REM DAL                              <012>
60475 PRINT,"{BROWN,SPACE}@@@@@@@@@@@@@@@@ <169>
60480 PRINT,"R{SPACE,WHITE}.{6SPACE}.{4SPA
      CE,BROWN,3SPACE}G                    <061>
60485 PRINT,"R{SPACE,WHITE,4SPACE}.{5SPACE
      }. .{BROWN,2SPACE}G                  <096>
60490 PRINT,"R{SPACE,GREY 1,6SPACE}NM{BROW
      N,7SPACE}G                           <218>
60495 PRINT,"R{GREY 1,2SPACE}NM{2SPACE}N{2
      SPACE}MNM{BROWN,2SPACE,SHIFT-SPACE,S
      PACE}G                               <240>
60500 PRINT,"R{GREY 1,SPACE}N{2SPACE}MN{3S
      PACE}N{2SPACE}M{2SPACE}N{BROWN}G     <112>
60505 PRINT,"R{GREY 1}N{3SPACE}N{YELLOW,SP
      ACE}B{GREY 1,SPACE}N{4SPACE}MN{BROWN
      ,SPACE}G                             <155>
60510 PRINT,"R{YELLOW,3SPACE}B{SPACE,RVSON
      }U{SPACE,RVOFF,SPACE}RRD{SPACE,GREY
      1,SPACE}N{BROWN,2SPACE}G             <198>
60515 PRINT,"R{YELLOW,3SPACE,RVSON,SPACE,R
      VOFF}RRR{SPACE,RVSON,SPACE,RVOFF}RRR
      {SPACE,BROWN,3SPACE}G                <062>
60520 PRINT,"R{YELLOW,2SPACE,RVSON}V{4SPAC
      E,RVOFF}G{RVSON,4SPACE,RVOFF}T{SHIFT
      -SPACE,BROWN,2SPACE}G                <059>
60525 PRINT,"R{ORANGE,RVSON}TTTTTTTTTTTTT
      TT{RVOFF,BROWN}G                     <215>
60530 PRINT,"R{ORANGE,RVSON,16SPACE,RVOFF,
      BROWN}G":GOTO 63000                  <063>
62000 PRINT"{CLR}":PRINT,"{YELLOW,5SPACE}M
      MN N                                 <035>
62001 PRINT,"{2SPACE}M M{2SPACE}MN{2SPACE}
      N N                                  <183>
62002 PRINT," M{2SPACE}M M{2SPACE}N N{2SPA
      CE}N                                 <184>
62003 PRINT," M MM MN N NN N               <171>
62004 PRINT,"MMMM MMUINN NNNN              <150>
62005 PRINT,"MMMM*{RVSON}@@@@@@{RVOFF}£NNN
      N                                    <196>
62006 PRINT," MMM{RED,RVSON}£{6SPACE}*{RVO
      FF,YELLOW}NNN                        <222>
62007 PRINT," MM{RED,RVSON}£{8SPACE}*{RVOF
      F,YELLOW}NN                          <044>
62008 PRINT,"{2SPACE}M{RVSON}RN M{2SPACE}N
      RR{RVOFF}N                           <139>
62009 PRINT,"{2SPACE}M{RVSON}RN{2SPACE}OP{
      2SPACE}RR{RVOFF}N                    <152>
62010 PRINT,"{3SPACE,RVSON}RN{2SPACE}RN{2S
      PACE}RR{RVOFF}                       <049>
62011 PRINT,"{ORANGE,RVSON,3SPACE,YELLOW}R
      N{2SPACE}LG{2SPACE}RR{ORANGE,3SPACE,
      RVOFF}                               <154>
62012 PRINT,"{RVSON,3SPACE,YELLOW}RN N{2SP
      ACE}M RR{ORANGE,3SPACE,RVOFF}        <008>
62013 PRINT,"{RVSON,3SPACE}EKIM-REPUS{3SPA
      CE,RVOFF}                            <090>
62020 PRINT,"{4LEFT,DOWN,WHITE}HERZLICHEN
      GLUECKWUNSCH ,",","{4LEFT}SIE HABEN E
      S GESCHAFFT !                        <241>
62025 X$="{40SPACE}*** J.R.R TOLKIEN"      <007>
62030 X$=X$+" *** DER KLEINE HOBBIT *** EI
      N ABENTEUERSPIEL VON "               <254>
62040 X$=X$+"MICHAEL NICKLES - BERGSTR.3 -
      8763 KLINGENBERG 2 ***"              <138>
62070 X$=X$+"{41SPACE}"                    <215>
62100 FOR I=1 TO LEN(X$)-39                <067>
62110 POKE 211,1:POKE 214,20:SYS 58732     <229>
62120 PRINT MID$(X$,I,39)                  <189>
62130 FOR II=1 TO 150:NEXT II              <062>
62140 NEXT I:END                           <168>
62500 POKE 53280,0:POKE 53281,0:PRINT"{CLR
      ,RED,4DOWN}","{2SPACE}J.R.R. TOLKIEN
      "                                    <135>
62501 PRINT"{3DOWN}":PRINT,"{YELLOW,2SPACE
      }DER KLEINE                          <244>
62502 PRINT                                <117>
62503 PRINT,"{2SPACE}RN NM OP OP RPY       <127>
62504 PRINT,"{2SPACE}LG RN LG LG RN        <025>
62505 PRINT,"{2SPACE}RN RN RN RN RN        <225>
62506 PRINT,"{2SPACE}RN MN LG LG RN        <196>
62510 PRINT"{4DOWN,SPACE}EIN ABENTEUERSPIE
      L VON MICHAEL NICKLES                <222>
62520 PRINT,"{2DOWN,RED,2SPACE}BITTE WARTE
      N !"                                 <062>
62600 RETURN                               <171>
63000 POKE 198,0:WAIT 198,1:PRINT"{WHITE}"
      :RETURN                              <124>
63100 PRINT,"{SPACE,ORANGE}@@@@@@@@@@@@@@  <158>
63102 PRINT,"R{WHITE}.{5SPACE}.{4SPACE}.{S
      PACE,ORANGE,SPACE}G                  <093>
63103 PRINT,"R{WHITE,3SPACE}.{5SPACE}.{3SP
      ACE}.{ORANGE}G                       <094>
63104 PRINT,"R{14SPACE}G                   <179>
63105 PRINT,"R{6SPACE,WHITE,RVSON}£*{RVOFF
      ,ORANGE,6SPACE}G                     <173>
63106 PRINT,"R{5SPACE,WHITE,RVSON}£{2SPACE
      }*{RVOFF,ORANGE,5SPACE}G             <174>
63107 PRINT,"R{4SPACE,GREY 1,RVSON}£{4SPAC
      E}*{RVOFF,ORANGE,4SPACE}G            <083>
63108 PRINT,"R{3SPACE,GREY 1,RVSON}£{6SPAC
      E}*{RVOFF,ORANGE,3SPACE}G            <084>
63109 PRINT,"R{2SPACE,GREY 1,RVSON}£{8SPAC
      E}*{RVOFF,ORANGE,2SPACE}G            <085>
63110 PRINT,"R{RVSON}TTTTTTTTTTTTTT{RVOFF}
      G                                    <043>
63111 PRINT,"R{RVSON,14SPACE,RVOFF}G":GOTO
      63000                                <088>
```

**Listing: »Der kleine Hobbit« (Schluß)**

# Cohan's Land

**Ergründen Sie das Geheimnis der Nachricht aus der Flaschenpost. Ist der Schatz wirklich vorhanden oder handelt es sich nur um einen Scherz?**

Cohan's Land ist ein deutsches Textadventure, das ich vor etwa einem Jahr geschrieben habe. Die Entstehungsgeschichte ist ziemlich interessant: Als ich das Manuskript zu meinem ersten Adventure-Programmierkurs schrieb, stellte ich im Rahmen dieses Kurses vor, wie man ein Adventure schreibt. Also wie man eine Idee aufgreift und diese dann ausarbeitet. Dabei kam mir die Idee zu Cohan's Land. Cohan's Land ist ein Adventure, das im Scott Adams-Stil programmiert ist. Scott Adams, ein Pionier der Abenteuerspielprogrammierung, hat Spiele wie »Adventureland«, »Pirate Adventure«, »the Count«, »Voodoo Castle« sowie die Questprobe-Serie »the Hulk«, »Spiderman« etc. entworfen. Erstaunlich ist die Tatsache, daß Scott Adams sein absolut miserables Zwei-Wort-Parsersystem auch heute noch verwendet. Seine Adventures verstehen nur Befehle, die sich maximal aus zwei Worten zusammensetzen (TAKE SWORD). Auf großen Textumfang legt er auch keinen Wert – die Texte beschreiben nur das Wesentlichste und sind so knapp wie möglich gehalten. Fast der gesamte zur Verfügung stehende Speicherplatz wird bei Adams für die Grafik verbraten, die sich komplett auf einmal im Speicher befindet – es wird nie nachgeladen.

Trotz des schlechten Programmierkonzeptes haben diese Adventures jedoch etwas Einzigartiges an sich: Der Spieler wird knallhart mit Problemen und Denknüssen konfrontiert. Man muß nicht lange herumlaufen und schöne Raumbeschreibungen lesen wie bei INFOCOM – es geht sofort richtig rund. So auch bei Cohan's Land. Die Texte sind knapp gehalten, die Probleme hingegen äußerst knifflig. Der Parser ist jedoch wesentlich besser ausgefallen. Der Wortschatz umfaßt etwa 140 Wörter und der Parser versteht ganze Sätze. Trotzdem ist das Programm nur knapp 30 KByte lang.

Die Story ist äußerst interessant: Im Spiel tauchen sowohl Fantasy-, Science-Fiction- und auch Nostalgie-Elemente auf. Der Spieler muß einen gefährlichen Dschungel überleben, Roboter bekämpfen und letztendlich die geheimnisvolle Quelle der Macht zerstören. Es fängt ganz harmlos mit einer Seereise an. Ein Forscher hat in einer Flaschenpost eine uralte Seekarte entdeckt, auf der eine bisher unbekannte Insel eingezeichnet ist. Die Expeditionsflotte geht bei einem gewaltigen Seesturm unter. Natürlich überlebt der Forscher und wird als Schiffbrüchiger auf die geheimnisvolle Insel verschlagen. Auf dieser Insel entdeckt er ein unvorstellbares Geheimnis...

(Michael Nickles/dm)

```
Gemuetlich eingerichtete Wohnstube.
In einem offenen Kamin knistert ein be-
hagliches Feuer.
Alter Schrank aus Eichenholz.
Haustuer. Kamin. Flasche.
Moegliche Richtungen : keine

Was soll ich nun machen ?

HILFE.
Was soll ich nun machen ?
```

Sie müssen sich irren. Unsere Computerspiele sind absolut ungefährlich!!?

**Listing »Cohan's Land«**

```
0 REM ************************************   <131>
2 REM *    C O H A N ' S   L A N D       *   <174>
4 REM *       COPYRIGHT 1984 BY          *   <246>
5 REM * SUPERMIKE'S FANTASY PRODUCTS     *   <023>
7 REM ************************************   <138>
8 POKE 53280,0:POKE 53281,0:PRINT CHR$(14)
  :SYS 65499:PRINT"{CLR,YELLOW}              <208>
9 ZN=1:RO=10000+ZN*100                       <137>
11 DATA 76,24,1,177,251,145,251,200,208,24
   9,230,252,202,208,244,96,120,160          <235>
12 DATA 0,169,160,132,251,133,252,162,32,3
   2,11,1,169,224,132,251,133,252            <065>
13 DATA 162,32,32,11,1,169,53,133,1,88,96:
   FOR I=264 TO 310:READ A:POKE I,A:NEXT     <141>
15 SYS 264:A=710:FOR I=A TO A+20:READ X:PO
   KE I,X:NEXT                               <130>
16 DATA 208,3,76,29,168,32,192,2,32,19,166
   ,56,165,95,233,1,164,96,76,36,168         <142>
18 POKE 40996,197:POKE 40997,2:POKE 1,54:A
   =43168:FOR I=A TO A+2:READ X:POKE I,X:N
   EXT                                       <047>
20 A=704:FOR I=A TO A+5:READ X:POKE I,X:NE
   XT:DATA 32,192,2,32,138,173,76,247,183    <074>
25 GOSUB 51000                               <053>
30 PRINT"{CLR,6DOWN}";CHR$(142)              <205>
36 PRINT,"{YELLOW}NT NPNPG NPNM GN NT        <010>
37 PRINT,"LPMPGN N OPN MG{2SPACE}YP          <133>
38 PRINT,"{17SPACE}TT                        <107>
39 PRINT,"{4SPACE}H{2SPACE}NPNM HOM          <050>
40 PRINT,"{4SPACE}LF OPN MHLN                <034>
42 PRINT,"{3DOWN,LIG.RED,LEFT}EIN ABENTEUE
   RSPIEL VON                                <049>
43 PRINT,"{DOWN,LIG.RED,2SPACE}MICHAEL NIC
```

```
     KLES":POKE 198,0:WAIT 198,1                  <225>
  50 PRINT"{CLR}";CHR$(14)                        <122>
  55 REM GE$(14)="DRAHTGITTER":GE(15)=-1:MA=
     0:GE(10)=-1:GE(16)=-1:GE(6)=-1              <134>
  56 REM    GE(1)=-1:ZB=10                        <025>
  57 REM FORI=1TOGZ:GE(I)=-1:NEXT                 <192>
  60 GOTO 1040                                    <128>
1000 REM ACTIONMODUL --------------------         <024>
1001 IF RI THEN 1015                              <054>
1002 GOSUB 50000                                  <252>
1010 IF RI=0 THEN 2000                            <084>
1015 IF RI(RI)=0 THEN PRINT"KEIN WEG IN DI
     ESE RICHTUNG !":RI=0:GOTO 2000              <205>
1016 VE=0:OM=0:OB=0:PE=0:PM=0:REM GE=0            <171>
1020 RO=10000+100*RI(RI):ZN=RI(RI):RI=0           <188>
1040 RESTORE RO:FOR I=1 TO 10:READ RI(I):N
     EXT                                          <152>
1041 IF PA(7)AND ZN=37 THEN RI(2)=38              <232>
1050 PRINT"{CLR,YELLOW}";:GOSUB ROOM              <127>
1100 REM ***** SIE SEHEN                          <210>
1700 SE$="":IC=0:FOR I=1 TO PZ:IF PE(I)=ZN
      THEN PRINT PE$(I)". ";:IC=1                 <245>
1710 NEXT:IF IC THEN PRINT                        <178>
1820 FOR I=1 TO OZ:IF OO(I)=ZN THEN SE$=SE
     $+O$(I)+". ":IC=1                            <218>
1822 NEXT                                         <054>
1830 FOR I=1 TO GZ:IF GE(I)=ZN THEN SE$=SE
     $+GE$(I)+". ":IC=1                           <070>
1831 IF GE(I)=-2 AND ZN=1 AND TU(1)=0 THEN
      SE$=SE$+GE$(I)+". ":IC=1                    <015>
1832 IF GE(I)=-3 AND ZN=5 AND TU(3)=0 THEN
      SE$=SE$+GE$(I)+". ":IC=1                    <054>
1833 IF GE(I)=-4 AND ZN=43 AND TU(4)=0 THE
     N SE$=SE$+GE$(I)+". ":IC=1                   <110>
1834 NEXT I                                       <140>
1835 IF SE$=""THEN 1900                           <157>
1840 IF LEN(SE$)<41 THEN PRINT SE$:GOTO 19
     00                                           <209>
1850 I$=LEFT$(SE$,39)                             <019>
1860 IM=0:FOR I=1 TO LEN(I$)                      <197>
1865 :IF MID$(I$,I,1)=" "THEN IM=I                <191>
1868 NEXT I                                       <174>
1870 I$=LEFT$(I$,IM):PRINT I$                     <166>
1880 SE$=RIGHT$(SE$,LEN(SE$)-IM)                  <231>
1890 GOTO 1840                                    <052>
1900 PRINT"MOEGLICHE RICHTUNGEN : ";              <147>
1910 IC=0:FOR I=1 TO 10:IF RI(I)THEN PRINT
      RI$(I)",";:IC=1                             <212>
1920 NEXT:IF IC=0 THEN PRINT"KEINE ";             <179>
1930 PRINT"{LEFT,SPACE,DOWN}"                     <146>
2000 REM REAGIEREN AUF BEFEHLE ---------          <090>
2005 REM FEHLERMELDUNGEN ***************          <073>
2010 IF OB AND OM=0 THEN PRINT"DAS GEHT HI
     ER NICHT .... NOCH NICHT !"                  <205>
2015 IF PE AND PM=0 THEN PRINT"ICH SEHE DI
     ESE PERSON HIER NICHT !"                     <135>
2019 IF VE=16 THEN 2100                           <053>
2020 IF VE<>5 AND VE<>8 AND G1 AND GM=0 AN
     D G1<>12 THEN PRINT"ICH HABE DAS NICH
     T !"                                         <153>
2100 REM EINWORT-BEFEHLE ***************          <040>
2105 IF VE=3 AND OB=0 AND GM=0 THEN VE=0:G
     OTO 1050:REM LOOK                            <192>
2110 IF VE<>12 THEN 2130:REM INVENTUR             <176>
2111 SE$="ICH HABE: ":IC=0:FOR I=1 TO GZ:I
     F GE(I)=-1 THEN SE$=SE$+GE$(I)+", ":I
     C=1                                          <205>
2112 NEXT I:IF IC=0 THEN SE$=SE$+" NICHTS.
     "                                            <046>
2113 SE$=LEFT$(SE$,LEN(SE$)-1)                    <106>
2116 IF LEN(SE$)<41 THEN PRINT SE$:GOTO 21
     30                                           <207>
2117 I$=LEFT$(SE$,39)                             <032>
2118 IM=0:FOR I=1 TO LEN(I$)                      <201>
2119 :IF MID$(I$,I,1)=" "THEN IM=I                <191>
2120 NEXT I                                       <172>
2121 I$=LEFT$(I$,IM):PRINT I$                     <163>
2122 SE$=RIGHT$(SE$,LEN(SE$)-IM)                  <219>
2123 GOTO 2116                                    <216>
2130 IF VE=19 THEN PRINT"ZEIT VERGEHT.....
     .."                                          <130>
2200 REM NIMM/VERLIER ROUT. ***********           <044>
2205 IF VE<>5 AND VE<>8 OR G1=0 OR OB THEN
      2237                                        <232>
2220 IF VE=5 AND GE(G1)=-1 THEN PRINT"ICH
     HABE DAS BEREITS !":GOTO 2300                <137>
2222 IF VE=8 AND GE(G1)<>-1 THEN PRINT"ICH
     HABE DAS NICHT !"                            <230>
2229 IF ZN=27 AND VE=5 AND GE(5)=-4 THEN G
     E(5)=-1:PRINT"OK.":GOTO 2300                 <255>
2230 IF VE=5 AND GE(G1)=ZN THEN PRINT"OK."
     :GE(G1)=-1:GOTO 2250                         <175>
2231 IF VE=8 AND GE(G1)=-1 THEN PRINT"OK."
     :GE(G1)=ZN                                   <227>
2232 IF VE=5 AND GE(G1)=-2 AND TU(1)=0 THE
     N PRINT"OK.":GE(G1)=-1:GOTO 2250             <018>
2233 IF VE=5 AND GE(G1)=-3 AND TU(3)=0 THE
     N PRINT"OK.":GE(G1)=-1:GOTO 2250             <084>
2234 IF VE=5 AND GE(G1)=-4 AND TU(4)=0 THE
     N PRINT"OK.":GE(G1)=-1:GOTO 2250             <118>
2237 IF G1 AND VE=5 AND GE(G1)<>ZN THEN PR
     INT"ICH SEHE DAS HIER NICHT !"               <017>
2238 IF VE=5 AND G1=0 AND OB<>0 THEN PRINT
     "DAZU BIN ICH NICHT STARK GENUG !"           <211>
2250 IF G1=15 AND FS=1 THEN FS=0                  <001>
2252 IF G1=14 AND FA+FS=2 THEN 2625               <002>
2254 IF VE=5 AND G1=16 THEN PRINT"ICH ZIEH
     E DEN SCHUTZANZUG AM BESTEN SO- FORT
     AN !"                                        <144>
2300 REM TOETE ************************           <054>
2310 IF VE<>20 THEN 2400                          <056>
2315 IF G1=0 THEN PRINT"WIE ? DOCH NICHT M
     IT MEINEN BLOSSEN{5SPACE}HAENDEN !":G
     OTO 2400                                     <185>
2320 IF GM=3 THEN PRINT"ICH GREIFE MIT DEM
      SAEBEL AN !"                                <242>
2322 IF GM=10 THEN PRINT"ICH GREIFE MIT DE
     R AXT AN !"                                  <179>
2324 IF GM=6 AND MU=0 THEN PRINT"ICH HABE
     KEINE MUNITION MEHR !":GE(7)=0:GOTO 2
     400                                          <090>
2326 IF GM=6 AND MU THEN PRINT"ICH SCHIESS
     E MIT DER PISTOLE !":MU=MU-1                 <006>
2330 IF PA(3)<6 THEN 2340                         <104>
2332 IF ZN<>71 OR GM<>6 THEN 2340                 <098>
2334 PRINT"DER LOEWE BRICHT TOT ZUSAMMEN."
     :PE$(3)="TOTER LOEWE":PA(3)=-1               <214>
2335 PE(3)=10:SC=SC+7.5                           <178>
2340 IF GM<>6 AND GM<>3 AND GM<>10 OR PA(4
     )<1 THEN 2350                                <239>
2342 PRINT"DAS KROKODIL STUERZT BLUTEND IN
     S WASSER.":PA(4)=-1:SC=SC+3.5                <255>
2350 IF ZN<>61 OR OM<>39 OR GM<>6 THEN 236
     0                                            <133>
2352 I=INT(100*RND(1))+1                          <128>
2354 IF I>50 OR BR=2 THEN PRINT"DIE KUGEL
     VERFEHLT DEN DRUCKKNOPF KNAPP.":BR=-1        <098>
2356 IF I<=50 AND BR<>2 THEN PRINT"DIE KUG
     EL TRIFFT DEN DRUCKKNOPF EXAKT !":BR=
     1                                            <174>
2360 :                                            <050>
2400 REM BAUE FLOSS ******************            <166>
2410 IF VE<>25 OR G1<>12 THEN 2500                <032>
2420 IF GE(11)<>-1 THEN PRINT"ICH HABE KEI
     N HOLZ !":GOTO 2500                          <011>
2430 PRINT"OK.":GE(11)=0:GE(12)=-1                <130>
2500 IF VE<>31 AND VE<>42 THEN 2600: REM V
     ERBINDE ******************                   <029>
2505 IF G1+G2<>29 THEN 2600                       <209>
2510 IF GE(14)<>-1 THEN PRINT"ICH HABE DAS
      DRAHTGITTER NICHT !":GOTO 2600              <247>
2520 IF GE(15)<>-1 THEN 2600                      <130>
2530 IF FA=1 THEN PRINT"DAS HABE ICH BEREI
     TS.":GOTO 2600                               <151>
2535 PRINT"OK.":FA=1:REM IFFS=1THEN2625           <121>
2600 IF VE<>44 AND VE<>42 THEN 2700:REM ST
     ECKE **************                          <247>
2610 IF OM<>26 THEN 2700                          <204>
2620 IF FS=0 THEN PRINT"OK.":GE(15)=ZN:FS=
     1                                            <073>
2622 IF FA<>1 OR GE(14)<>-1 THEN 2630             <102>
2625 PRINT"DAS DRAHTGITTER GLUEHT AUF UND
     VERABREI-CHT MIR EINEN ELEKTROSCHOCK"
     ;                                            <123>
2626 PRINT" VON MEHRERENTAUSEND VOLT.":GOT
     O 8000                                       <151>
2630 IF FS=0 AND GE(15)=-1 THEN PRINT"OK."
     :GE(15)=ZN:FS=1                              <116>
2700 REM SPEZIAL ACTION SCENARIO                  <118>
2710 IF ZN=43 OR ZN=51 THEN IF MA<>0 THEN
     6145                                         <199>
2715 IF GE(18)=-1 AND ZB=11 THEN PRINT"DIE
      ZEITBOMBE BEGINNT PLOETZLICH ZU{5SPA
     CE}TICKEN !"                                 <041>
```

**Listing »Cohan's Land« (Fortsetzung)**

```
2716 IF ZB<11 AND GE(18)=-1 THEN PRINT"DIE
     ZEITBOMBE TICKT IMMER SCHNELLER !"      <190>
2720 IF ZB THEN ZB=ZB-1:IF ZB=0 THEN PRINT
     "DIE ZEITBOMBE EXPLODIERT.":GOTO 8000   <162>
2730 I=PA(9): REM STRAHLENVERSEUCHUNG         <010>
2735 IF I=1 THEN PRINT"MIR WIRD GANZ KOMIS
     CH VON DEM VIOLETTEN LICHT !"           <050>
2740 IF I THEN PA(9)=PA(9)+1:I=PA(9)          <092>
2745 IF I=5 THEN PRINT"ICH WERDE PLOETZLIC
     H GANZ SCHWACH !"                       <226>
2746 IF I=8 THEN PRINT"ICH WERDE IMMER SCH
     WAECHER !"                              <206>
2747 IF I=10 THEN PRINT"MEINE KRAEFTE VERL
     ASSEN MICH .....":GOTO 8000             <098>
2800 IF VE=28 AND GM=1 THEN PRINT"ICH BIN
     KEIN SCHIFFSNAVIGATOR !"                <225>
2810 IF VE<>37 OR GM<>8 OR GE$(8)="GLASSCH
     ERBEN"THEN 2820                         <182>
2815 PRINT"DIE FLASCHE ZERSPRINGT IN GLASS
     CHERBEN. EINE SEEKARTE FAELLT AUF ";    <161>
2816 PRINT"DEN BODEN.":GE$(8)="GLASSCHERBE
     N":GE(8)=ZN:GE(1)=ZN:SC=SC+2.5          <037>
2820 IF VE<>49 THEN 2830                      <148>
2822 IF GE(2)<>-1 AND GE(4)<>-1 THEN PRINT
     "ICH HABE KEINEN HUNGER.":GOTO 2830     <056>
2824 PRINT"OK.":GE(2)=0:GE(4)=0:SC=SC+2.5     <132>
2830 :                                        <012>
4000 REM HILFE ************************       <112>
4010 IF VE<>30 THEN 5000                      <084>
4999 REM  GOSUB61020                          <229>
5000 REM U N T E R S U C H E -----------      <160>
5001 IF VE<>3 OR OM=0 AND GM=0 THEN 5999      <070>
5010 IC=0:IF OM=7 THEN PRINT"DIE MAUER IST
     UNUEBERWINDBAR HOCH.":IC=1              <077>
5015 IF OM=14 AND OO(16)<>22 THEN SC=SC+6.
     5                                       <252>
5020 IF OM=14 THEN PRINT"EINE SCHRIFT IST
     IN DIE HAND GERITZT.":OO(16)=22:IC=1    <011>
5030 IF OM<>17 THEN 5035                      <118>
5031 PRINT"DIE RECHTE HAND DES SKELETTS UM
     KLAMMERT EINEN SPITZEN KREIDESTEIN."    <097>
5032 IC=1:OO(15)=22                           <044>
5035 IF OM<>20 THEN 5037                      <053>
5036 PRINT"IN EINEM TROPFSTEIN IST EINGERI
     TZT :{4SPACE}POWERPETER{SHIFT-SPACE}W
     AR HIER !":IC=1                         <119>
5037 IF OM=24 OR OM=1 THEN VE=0:GOTO 1050     <101>
5040 IF GM=7 THEN PRINT"ICH HABE NOCH";MU;
     "SCHUSS !":IC=1                         <242>
5050 IF GM<>15 THEN 5055                      <104>
5051 PRINT"DAS KABEL IST CA 50M LANG. AM E
     INEN ENDEHAT ES EINEN STECKER, AM ";    <089>
5052 PRINT"ANDEREN ENDE{3SPACE}IST ES BLAN
     K."                                     <138>
5053 IC=1                                     <115>
5055 IF OM<>37 OR GE(17)<>0 THEN 5060         <213>
5056 PRINT"DER TOTE HAT EINEN SCHLUESSEL U
     ND EINE{2SPACE}AKTIVIERTE ZEITBOMBE "
     ;                                       <206>
5057 PRINT"BEI SICH !":GE(17)=ZN:GE(18)=ZN
     :IC=1:ZB=50                             <083>
5060 IF GM<>18 THEN 5065                      <133>
5062 PRINT"AUF DER ZEITBOMBE IST EIN ZAEHL
     ER ANGE- BRACHT DER MOMENTAN DEN ";     <092>
5063 PRINT"WERT";ZB;"HAT !":IC=1              <050>
5065 IF OM<>43 THEN 5070                      <183>
5066 PRINT"AM BODEN DES BRUNNENS STRAHLT E
     IN HELLESLICHT-DIE QUELLE{SHIFT-SPACE
     }DER{SHIFT-SPACE}MACHT !"               <229>
5067 IC=1                                     <129>
5070 IF GM=8 THEN PRINT"IN DER FLASCHE IST
     EINE SCHRIFTROLLE.":IC=1                <115>
5075 IF GM=16 THEN PRINT"DER ANZUG SCHUEZT
     VOR RADIOAKTIVER{6SPACE}STRAHLUNG.":
     IC=1                                    <192>
5500 IF IC=0 THEN PRINT"ICH ENTDECKE NICHT
     S BESONDERES."                          <157>
5999 :                                        <133>
6000 REM NICHTSPIELERPERSONEN ----------      <126>
6010 REM ********** LOEWE                     <181>
6015 IF ZN=71 AND PA(3)<>-1 THEN 6035         <015>
6020 IF ZN<6 OR ZN>13 OR PA(3)=-1 OR INT(1
     0*RND(1))+1<6 OR PA(4)THEN 6100         <089>
6035 PA(3)=PA(3)+1:IF PA(3)<6 THEN PRINT"I
     M GEBUESCH RASCHELT ETWAS !":REM  610
     0                                       <059>
6050 IF PA(3)=6 THEN PRINT"EIN RIESSIGER L
     OEWE SPRINGT AUS DEM GE- BUESCH HERVO
     R !"                                    <134>
6051 IF PA(3)=6 THEN PE(3)=ZN                 <144>
6055 IF PA(3)=7 AND ZN=71 THEN PRINT"DER L
     OEWE SPRINGT AM BAUMSTAMM HOCH !"       <049>
6060 IF PA(3)=8 THEN PRINT"DER LOEWE FAELL
     T MICH AN !":GOTO 8000                  <051>
6100 REM ********** ROBOTER                   <012>
6105 IF ZN<38 OR MA=0 THEN 6200               <090>
6110 MW=MW+1:MA=MA(MW):IF MA=-1 THEN MW=0:
     GOTO 6110                               <221>
6130 IC=0:FOR I=1 TO 10:IF RI(I)=MA THEN I
     C=1                                     <120>
6135 NEXT:IF IC THEN PRINT"ICH HOERE SCHRI
     TTE IN DER NAEHE !"                     <251>
6136 IF MA<>GE(14)OR FA+FS<>2 THEN 6140       <008>
6137 MA=0:PE(4)=GE(14):SC=SC+7                <231>
6138 IF PE(4)<>ZN THEN PRINT"ICH HOERE EIN
     LAUTES KRACHEN !":GOTO 6140             <195>
6139 PRINT"DER ROBOTER LAUEFT UEBER DAS DR
     AHTGITTERUND VERSCHMORT !"              <188>
6140 IF MA<>ZN THEN 6150                      <060>
6145 PRINT"EIN RIESSIGER ROBOTER ERSCHEINT
     !"                                      <149>
6146 PRINT"AUS DEN AUGEN DES ROBOTERS SCHI
     ESST EIN LICHTSTRAHL AUF MICH ....."    <216>
6147 PRINT"ICH WERDE MUEDE ..... SEHR MUED
     E ....."                                <075>
6148 RI(1)=39:RI=1:PA(8)=1:POKE 198,0:WAIT
     198,1                                   <060>
6150 :                                        <030>
6200 GOSUB ROOM+20:GOTO 1000                  <184>
7000 REM VERLOREN                             <235>
8000 PRINT"ICH BIN TOT."                      <110>
8010 PRINT"{WHITE}SIE HABEN";SC;"% DIESES
     ADVENTURES GELOEST."                    <085>
8020 FOR I=1 TO 3000:NEXT:POKE 198,0:WAIT
     198,1:RUN                               <134>
10000 REM DIE LANDKARTE -----------------     <178>
10100 DATA,,,,,,,,,,,:PRINT"GEMUETLICH EIN
      GERICHTETE WOHNSTUBE."                 <094>
10102 PRINT"IN EINEM OFFENEN KAMIN KNISTER
      T EIN BE- HAGLICHES FEUER."            <018>
10109 IF TU(2)=0 THEN RI(1)=2                 <230>
10120 IF VE=1 AND OM=1 THEN PRINT"OK.":TU(
      1)=0                                   <186>
10125 IF VE=2 AND OM=1 THEN PRINT"OK.":TU(
      1)=1                                   <003>
10127 IF VE=1 AND OM=2 OR OM=5 THEN PRINT"
      OK.":TU(2)=0:RI(1)=2                   <047>
10128 IF VE=2 AND OM=2 OR OM=5 THEN PRINT"
      OK.":TU(2)=1:RI(1)=0                   <239>
10129 IF VE<>8 OR OM<>1 THEN 10140            <214>
10130 IF TU(1)THEN PRINT"DER SCHRANK IST G
      ESCHLOSSEN !":GOTO 10140               <009>
10131 IF GE(G1)=-1 THEN PRINT"OK.":GE(G1)=
      -2:REM GOTO                            <131>
10140 IF VE=8 AND OM=10 AND GM=1 THEN PRIN
      T"DIE SEEKARTE VERBRENNT.":GE(1)=0     <232>
10150 RETURN                                  <046>
10200 DATA 3,,,,,,,,,,,:PRINT"BREITE STRAS
      SE DIE ZUM HAFEN FUEHRT."              <062>
10203 PRINT"MR BOLINGS HAUS.                  <101>
10205 IF TU(2)=0 THEN RI(2)=1                 <132>
10220 IF VE=1 AND OM=5 THEN PRINT"OK.":TU(
      2)=0:RI(2)=1                           <083>
10221 IF VE=2 AND OM=5 THEN PRINT"OK.":TU(
      2)=1:RI(2)=0                           <148>
10230 RETURN                                  <126>
10300 DATA,2,,,,,,,,:PRINT"AM GROSSEN HAFE
      N. EIN GROSSES HANDELS-{2SPACE}SCHIF
      F, ";                                  <127>
10305 PRINT"DIE 'QUEEN MARY' WIRD GERADE{4
      SPACE}BELADEN."                        <191>
10320 IF VE=13 AND OB=3 THEN RI(1)=4:RI=1:
      TI$="000000"                           <179>
10330 RETURN                                  <228>
10400 DATA,,,,,,,,,,:PRINT"AUF DER QUEEN M
      ARY."                                  <161>
10410 IF PA(0)THEN 10419                      <007>
10411 PRINT"KAEPTAN GRAUBART SAGT : 'HALLO
      , WILLKOM-MEN AN BORD MR BOLINGS ! "
      ;                                      <139>
10412 PRINT"NAVIGATOR AN{3SPACE}DECK, HIEV
      T DIE SEGEL, HOLT DEN ANKER{3SPACE}E
      IN!'{DOWN}"                            <168>
```

```
10413 PA(0)=1                                   <164>
10419 RETURN                                    <061>
10420 IF VE=13 AND OB=3 THEN RI(1)=3:RI=1       <070>
10424 PA(0)=PA(0)+1:IF PA(0)>3 THEN PA(0)=
      3                                         <089>
10425 IF PA(0)=3 THEN PRINT"{YELLOW}DAS SC
      HIFF BEFINDET SICH AUF HOHER SEE."        <068>
10426 IF TI$>"000300"THEN 10440                 <193>
10427 IF GE(1)<>-1 THEN 10430                   <082>
10428 IF VE+PE+G1=18 THEN PRINT"{DOWN}DER
      NAVIGATOR NIMMT DIE KARTE.":PA(1)=1:
      GE(1)=0                                   <179>
10429 IF VE+PE+G1=18 THEN SC=SC+2.5             <173>
10430 IF PA(1)THEN PA(1)=PA(1)+1                <190>
10431 IF PA(1)=4 THEN PRINT"EIN SCHIFFSJUN
      GE RUFT 'LAND IN SICHT !'.":TI$="000
      300"                                      <050>
10432 IF TI$>"000030"AND PA(2)=0 THEN PRIN
      T"EIN SCHWERER STURM BRICHT AUS !":P
      A(2)=1                                    <207>
10433 RETURN                                    <075>
10440 PRINT"EIN STARKER RUCK ZUCKT PLOETZL
      ICH DURCH DAS SCHIFF !"                   <218>
10441 FOR I=1 TO 4000:NEXT                      <070>
10445 PRINT"EIN MATROSE RUFT 'WIR SIND AUF
       EIN RIFF AUFGEFAHREN UND HABEN EIN
      ";                                        <012>
10446 PRINT"LECK......WIR SINKEN....."          <187>
10450 POKE 198,0:WAIT 198,1                     <006>
10460 IF PA(1)=0 THEN PRINT"ICH ERTRINKE..
      ..":GOTO 8000                             <039>
10470 RI(1)=5:RI=1:RETURN                       <118>
10500 DATA,,,,,,6,,,:PRINT"AM INSELSTRAND.
      "                                         <227>
10502 PRINT"IM WASSER IN STRANDNAEHE TREIB
      EN EINIGE SCHIFFSWRACKTEILE.":RETURN      <084>
10510 RETURN                                    <154>
10520 IF VE<>19 OR OO(4)THEN IF PA(3)<>5 T
      HEN 10523                                 <073>
10521 PRINT"EINE RIESSIGE HOLZTRUHE WIRD P
      LOETZLICH AN DEN STRAND GETRIEBEN.        <247>
10522 OO(4)=5:PA(3)=6                           <131>
10523 PA(3)=PA(3)+1                             <129>
10530 IF VE=1 AND OM=4 THEN PRINT"OK.":TU(
      3)=0:O$(4)="OFFENE TRUHE"                 <171>
10531 IF VE=2 AND OM=4 THEN PRINT"OK.":TU(
      3)=1:O$(4)="GROSSE SEEMANNSTRUHE"         <133>
10535 IF VE=8 AND OM=4 AND TU(3)=0 AND GE(
      61)=-1 THEN PRINT"OK.":GE(61)=-3          <033>
10540 RETURN                                    <184>
10600 DATA,13,,,,5,12,,,:PRINT"IN DICHTEM
      DSCHUNGEL."                               <052>
10620 RETURN                                    <008>
10700 DATA,10,,,8,,9,,,:PRINT"IN DICHTEM D
      SCHUNGEL."                                <177>
10720 RETURN                                    <108>
10800 DATA,9,,,,,,7,,:PRINT"IN DICHTEM DSC
      HUNGEL."                                  <220>
10820 IF VE=23 AND OM=7 THEN PRINT"DIE MAU
      ER IST VIEL ZU HOCH DAZU !"               <159>
10825 IF VE=8 AND GM=5 AND OM=7 THEN PRINT
      "DAS SEIL IST LEIDER ZU KURZ !"           <193>
10830 RETURN                                    <220>
10900 DATA 8,8,,10,,7,,,,:PRINT"IN DICHTEM
       DSCHUNGEL."                              <154>
10920 RETURN                                    <054>
11000 DATA,11,9,,,,,11,,:PRINT"OFFENE LICH
      TUNG IM DSCHUNGEL.":RETURN                <226>
11020 IF VE=23 AND OM=6 THEN RI(1)=71:RI=1      <060>
11030 RETURN                                    <166>
11100 DATA 10,,,,,12,,,,:PRINT"IN DICHTEM
      DSCHUNGEL."                               <055>
11120 IF VE=26 THEN PRINT"DIE KROKODILE LA
      SSEN DAS NICHT ZU !"                      <055>
11122 IF VE<>27 THEN 11134                      <080>
11124 IF GE(12)<>-1 THEN PRINT"WIE DENN ?"
      :GOTO 11150                               <243>
11130 IF PA(4)=0 AND GE(12)=-1 THEN PA(4)=
      1:PRINT"OK.ICH BEFINDE MICH AUF DEM
      FLUSS."                                   <219>
11132 RETURN                                    <012>
11134 IF PA(4)<>1 THEN 11138                    <171>
11135 PRINT"ZAHLREICHE KROKODILE KOMMEN HE
      RBEIGE-{3SPACE}SCHWOMMEN !"               <184>
11136 PRINT"EIN BESONDERS WILDES KROKODIL
      KLETTERT{2SPACE}AUF DAS KLEINE FLOSS
       !"                                       <160>
11137 PA(4)=2:RETURN                            <173>
11138 IF PA(4)=2 THEN PRINT"DAS KROKODIL K
      OMMT BEDROHLICH NAEHER !":PA(4)=3:RE
      TURN                                      <021>
11140 IF PA(4)=3 THEN PRINT"DAS KROKODIL H
      AT MICH GEFRESSEN.":GOTO 8000             <087>
11150 IF PA(4)=-1 THEN PRINT"DIE UEBRIGEN
      KROKODILE FRESSEN ES."                    <072>
11152 IF PA(4)=-1 THEN RI(1)=14:RI=1:PRINT
      "ICH HABE DAS ANDERE UFER ERREICHT."      <002>
11154 IF PA(4)=-1 THEN PRINT"DAS FLOSS TRE
      IBT AUF{SHIFT-SPACE}DEM FLUSS DAVON.
      ":GE(12)=0                                <164>
11155 IF PA(4)=-1 THEN POKE 198,0:WAIT 198
      ,1                                        <253>
11160 RETURN                                    <040>
11200 DATA,,,13,10,6,11,,,:PRINT"IN DICHTE
      M DSCHUNGEL."                             <199>
11205 PRINT"IN DER NAEHE RAUSCHT WASSER."       <099>
11220 IF VE<>24 OR OM<>11 OR GE(10)<>-1 OR
       GE(11)<>0 THEN 11230                     <218>
11225 PRINT"OK.":O$(11)="GEFAELLTER BAUM":
      GE(11)=ZN                                 <104>
11230 RETURN                                    <110>
11300 DATA 6,,12,,,,,,,:PRINT"IN DICHTEM D
      SCHUNGEL."                                <205>
11320 RETURN                                    <202>
11400 DATA,16,,17,,,,,,:PRINT"IN DICHTEM D
      SCHUNGEL."                                <101>
11420 RETURN                                    <046>
11600 DATA 14,,,17,,,,,,:PRINT"IN DICHTEM
      DSCHUNGEL"                                <061>
11620 RETURN                                    <248>
11700 DATA 14,,16,,,,,18,,:PRINT"IN DICHTE
      M DSCHUNGEL AM GEBIRGSRAND."              <040>
11720 RETURN                                    <092>
11800 DATA,19,,,17,,,,,:PRINT"SEHR STEILER
       GEBIRGSPFAD."                            <035>
11820 RETURN                                    <194>
11900 DATA 18,20,,,,,,,,:PRINT"SEHR STEILE
      R GEBIRGSPFAD."                           <202>
11920 RETURN                                    <038>
12000 DATA 19,23,,21,,,,,,:PRINT"IM HOHEN
      GEBIRGE."                                 <157>
12020 RETURN                                    <138>
12100 DATA 0,24,20,,,,23,,,:PRINT"IM HOHEN
       GEBIRGE."                                <062>
12120 RETURN                                    <240>
12200 DATA,25,,,,,,,,:PRINT"IN EINER TROCK
      ENEN HOEHLE."                             <065>
12220 IF VE<>28 OR OM<>16 THEN 12250            <028>
12224 PRINT"DIE SCHRIFT LAUTET :"               <230>
12225 PRINT"ICH BIN DEM GEHEIMNIS DER MACH
      T AUF DER SPUR......."                    <161>
12226 PRINT".. CODEWORT ....{SHIFT-SPACE}N
      AHOC...                                   <009>
12230 PRINT"ACHTUNG VOR DEN ROBO.....           <051>
12235 PRINT"....DR. GORDON....                  <234>
12240 PRINT"MEHR KANN MAN LEIDER NICHT ENT
      ZIFFERN !"                                <225>
12250 RETURN                                    <114>
12300 DATA 20,26,,,,21,25,,,:PRINT"IM HOHE
      N GEBIRGE."                               <032>
12320 RETURN                                    <077>
12400 DATA 21,,,,,,26,,,:PRINT"IM HOHEN GE
      BIRGE."                                   <228>
12420 RETURN                                    <030>
12500 DATA,,,26,,23,27,,,:PRINT"IM HOHEN G
      EBIRGE."                                  <177>
12520 IF VE=13 AND OB=13 THEN RI(1)=22:RI=
      1                                         <036>
12530 RETURN                                    <140>
12600 DATA 23,28,25,,,24,28,30,,:PRINT"IM
      HOHEN GEBIRGE."                           <100>
12620 RETURN                                    <232>
12700 DATA,,,,,25,,,,:PRINT"IM HOHEN GEBIR
      GE AUF EINEM FELSPLATEAU."                <235>
12710 PRINT"CA 10 METER UNTER MIR IST EIN
      SPALT IN{2SPACE}DER GEBIRGSWAND."         <136>
12715 IF GE(5)=-4 THEN PRINT"EIN AM RING B
      EFESTIGTES SEIL HAENGT HIN-AB."           <193>
12720 IF OM=18 AND GM=5 AND VE=31 THEN GE(
      5)=-4:PRINT"OK."                          <252>
12725 IF VE=23 AND GE(5)<>-4 THEN PRINT"DI
      E WAND IST ZU STEIL !"                    <160>
12730 IF VE=23 AND GE(5)=-4 THEN PRINT"OK.
      ":RI(1)=31:RI=1:GE(5)=31:SC=SC+8          <077>
12740 RETURN                                    <096>
12800 DATA,,,26,,26,,29,,:PRINT"IM HOHEN G
```

**Listing »Cohan's Land« (Fortsetzung)**

```
      EBIRGE."                               <125>
12820 RETURN                                 <178>
12900 DATA,,,,29,30,,,,:PRINT"IM HOHEN GEB
      IRGE."                                 <214>
12920 RETURN                                 <022>
13000 DATA,,,,26,,29,,,:PRINT"IM HOHEN GEB
      IRGE."                                 <252>
13020 RETURN                                 <122>
13100 DATA,,,,,,,,,,:PRINT"AUF EINEM FELSVO
      RSPRUNG AN EINER STEILEN";             <119>
13110 PRINT"GEBIRGSWAND."                    <121>
13120 IF VE=13 AND OM=19 THEN RI(1)=32:RI=
      1                                      <143>
13130 RETURN                                 <234>
13200 DATA 33,,31,,,,,,,,:PRINT"IN EINER GR
      OTTE."                                 <132>
13220 RETURN                                 <068>
13300 DATA,33,,34,,,,,,:PRINT"SCHMALER GAN
      G IN DER GROTTE."                      <240>
13320 RETURN                                 <170>
13400 DATA 36,,,35,,,,,,:PRINT"ENGER DURCH
      GANG."                                 <206>
13401 IF PA(5)THEN PRINT"DURCH STEINE VERS
      CHUETTETER AUSGANG."                   <004>
13405 IF PA(5)<>0 THEN 13420                 <081>
13406 PRINT"HINTER MIR BRICHT EINE STEINLA
      WINE AUS{2SPACE}UND VERSCHUETTET DEN
       AUS";                                 <251>
13407 PRINT"GANG ZUM SPALT{2SPACE}GAENZLIC
      H !":PA(5)=1                           <200>
13420 RETURN                                 <014>
13500 DATA,,34,,,,,,,,:PRINT"FEUCHTE GROTTE
      ."                                     <073>
13520 RETURN                                 <114>
13600 DATA,34,,,,,,,,,:PRINT"IN EINER GROTT
      E."                                    <206>
13601 PRINT"IN DER HAND IST EINE STAHLTUER
       EINGELAS-SEN.":IF PA(6)<>0 THEN 136
      20                                     <201>
13602 PRINT"HINTER DEN FELSEN SCHNELLEN ST
      AHLGREIF- ARME HERVOR UND HALTEN MI"
      ;                                      <110>
13603 PRINT"CH FEST."                        <056>
13604 PRINT"EIN KLEINER APPARAT SCHWEBT DI
      REKT VOR{2SPACE}MEIN GESICHT.":PA(6)
      =1                                     <211>
13605 RETURN                                 <201>
13620 IF PA(6)<>1 THEN 13640                 <239>
13621 PRINT"EINE SYNTHETISCHE STIMME AUS D
      EM APPARATFRAGT :";                    <043>
13622 PRINT"'WIE LAUTET DAS CODEWORT ?'.":
      GOSUB 50000                            <084>
13625 IF VE=34 THEN PA(6)=-1                 <218>
13630 IF PA(6)=1 THEN PRINT"EIN LASERSTRAH
      L SCHIESST AUS DEM APPARAT.":GOTO 80
      00                                     <070>
13632 PA(6)=-1:PRINT"DIE SYNTHETISCHE STIM
      ME SAGT : 'CODEWORTAKZEPTIERT !'."     <070>
13633 PRINT"DIE GREIFARME LASSEN MICH WIED
      ER FREI,{2SPACE}DER APPARAT SCHWEBT
      DAVON."                                <039>
13634 PRINT"DIE STAHLTUER OEFFNET SICH MIT
       EINEM{4SPACE}LEISEN SURREN.":OO(22)
      =36                                    <073>
13635 O$(21)="OFFENE STAHLTUER"              <248>
13640 IF VE=13 AND OM=22 THEN RI(1)=37:RI=
      1                                      <011>
13650 RETURN                                 <246>
13700 DATA,36,,,,,,,,,:PRINT"IN EINEM AUFZU
      G.":RETURN                             <159>
13720 IF VE<>35 OR OM<>23 OR PA(7)=3 THEN
      13730                                  <190>
13721 PRINT"DIE STAHLTUER SCHLIESST SICH M
      IT EINEM{2SPACE}LEISEN SURREN.         <161>
13722 IF PA(7)=0 THEN PRINT"DIE ROTE TASTE
       BLINKT KURZ HELL AUF."                <095>
13723 IF PA(7)=0 THEN PRINT"SELTSAM - DIE
      STAHLTUER OEFFNET SICH{4SPACE}WIEDER
       !"                                    <046>
13724 IF PA(7)=0 OR PA(7)=3 THEN 13730       <060>
13725 PRINT"DER AUFZUG BEWEGT SICH LEISE N
      ACH UNTEN.";:SC=SC+9.5                 <009>
13726 PRINT"NACH CA 2 MINUTEN BLEIBT ER ST
      EHEN !"                                <243>
13727 PRINT"DIE STAHLTUER OEFFNET SICH.":P
      A(7)=3:RI(2)=38                        <093>
13730 IF VE=1 AND OM=24 THEN PRINT"DER SCH
      ALTKASTEN IST VERRIEGELT !"            <027>
13735 IF VE<>37 OR OM<>24 THEN 13750         <253>
13737 IF GM=0 THEN PRINT"DOCH NICHT ETWA M
      IT MEINEN BLOSSEN HAEN-DEN !"          <029>
13739 IF GM=3 THEN PRINT"MIT DEM SAEBEL SC
      HAFFE ICH DAS NICHT !"                 <222>
13740 IF GM<>10 THEN 13750                   <205>
13741 PRINT"MIT LAUTEM KRACHEN FAELLT DIE
      BLECHAB-{2SPACE}DECKUNG DES SCHALTKA
      STENS";                                <036>
13742 PRINT" AUF DEN BODEN.":GE(13)=37       <213>
13743 OO(25)=37:O$(24)="AUFGEBROCHENER SCH
      ALTKASTEN"                             <017>
13750 IF VE=31 OR VE=42 THEN IF OM=25 THEN
       PRINT"OK.":PA(7)=1                    <104>
13751 IF PA(7)<>1 THEN 13760                 <243>
13752 PRINT"DIE ROTE TASTE LEUCHTET HELL A
      UF."                                   <042>
13753 O$(23)="HELL LEUCHTENDE ROTE TASTE":
      PA(7)=2                                <189>
13754 O$(25)="IM SCHALTKASTEN SIND ZWEI MI
      TEINANDER KURZGESCHLOSSENE ROTE KABE
      L"                                     <164>
13760 RETURN                                 <100>
13800 DATA,,42,,,,,,,,:PRINT"SCHWACH BELEUC
      HTETER RAUM."                          <000>
13820 IF VE=1 AND OM=33 THEN PRINT"ICH KAN
      N DIE STAHLTUER NICHT OEFFNEN."        <197>
13825 RETURN                                 <167>
13900 DATA,,,,,,,,,,:PRINT"RAUM MIT VERRIEG
      ELTER STAHLTUER.":RETURN               <065>
13920 IF PA(8)THEN PRINT"LANGSAM VERSCHWIN
      DET MEINE MUEDIGKEIT{3SPACE}WIEDER."   <113>
13921 IF PA(8)THEN PRINT"ICH BIN GEFANGEN
      !":PA(8)=0                             <086>
13925 IF VE<>13 OR OM<>27 THEN 13940         <057>
13930 IF GE(14)=0 THEN PRINT"EIN DRAHTGITT
      ER VERSCHLIESST DEN ZUGANG ZUM SCHAC
      HT."                                   <119>
13932 IF GE(14)<>0 THEN RI(1)=72:RI=1        <188>
13940 IF VE<>37 OR OM<>35 THEN 13950         <156>
13942 IF GM=0 THEN PRINT"WOMIT DENN ?":GOT
      O 13950                                <173>
13943 IF GM<>10 THEN 13950                   <186>
13944 PRINT"OK. DAS DRAHTGITTER FAELLT HER
      RUNTER UNDGIBT DEN EIN";:SC=SC+5       <058>
13945 PRINT"GANG ZUM SCHACHT FREI."          <231>
13946 OO(35)=0:GE(14)=39:GE$(14)="DRAHTGIT
      TER"                                   <157>
13950 RETURN                                 <036>
14000 DATA,41,51,,,,,,,,:PRINT"NORDENDE EIN
      ES KORRIDORS."                         <246>
14020 RETURN                                 <106>
14100 DATA 40,42,43,39,,,,,,:PRINT"LANGER
      KORRIDOR."                             <019>
14120 RETURN                                 <208>
14200 DATA 41,,44,38,,,,,,:PRINT"SUEDENDE
      EINES KORRIDORS."                      <134>
14220 RETURN                                 <052>
14243 PRINT"OK.":GE(GM)=-4                   <122>
14250 NEXT                                   <034>
14300 DATA,,,41,,,,,,,:PRINT"AUSSRUESTUNGSR
      AUM."                                  <113>
14320 IF VE=13 AND OM=29 THEN PRINT"DAZU B
      IN ICH ZU GROSS."                      <119>
14322 IF VE>2 OR OM<>29 THEN 14340           <168>
14325 IF TU(4)<>2 THEN 14330                 <119>
14326 IF GE(17)<>-1 THEN PRINT"DER SCHRANK
       IST VERRIEGELT !"                     <230>
14327 IF GE(17)<>-1 THEN PRINT"ICH HABE AU
      CH KEINEN SCHLUESSEL !"                <042>
14330 IF VE=1 AND GE(17)=-1 THEN PRINT"OK.
      ":TU(4)=0                              <196>
14335 IF VE=2 AND TU(4)=0 THEN PRINT"OK.":
      TU(4)=2                                <227>
14340 IF VE<>8 OR OM<>29 OR GM=0 THEN 1435
      0                                      <212>
14341 IF TU(5)=0 THEN PRINT"DER SCHRANK IS
      T NICHT OFFEN.":GOTO 14350             <232>
14343 PRINT"OK.":GE(GM)=-4                   <224>
14350 RETURN                                 <184>
14400 DATA,45,49,42,,,,,,:PRINT"RAUM."       <102>
14420 RETURN                                 <254>
14500 DATA 44,46,48,,,,,,,:PRINT"KORRIDOR.
      "                                      <160>
14520 RETURN                                 <098>
```

```
14600 DATA 45,,,,,,,,,:PRINT"ELEKTRIZITAET
      S-LABOR."                                 <168>
14620 RETURN                                    <200>
14700 DATA,,,,,,,,48,:PRINT"GROSSE HALLE."      <014>
14710 IF YT THEN YT=1:SC=SC+5                   <066>
14720 RETURN                                    <044>
14800 DATA,,,45,,,,,,47:PRINT"GEKACHELTER
      RAUM."                                    <083>
14820 RETURN                                    <144>
14900 DATA 50,,,44,,,,,,:PRINT"KLEINER KOR
      RIDOR."                                   <134>
14920 RETURN                                    <246>
15000 DATA,49,,,,,,,,,:PRINT"GROSSES ANTIK
      EINGERICHTETES WOHNZIMMER.";              <229>
15020 IF VE<>47 OR OM<>38 THEN RETURN           <172>
15021 IF YA THEN YA=1:SC=SC+5                   <192>
15022 PRINT"OK.":PRINT"DAS LICHT IM RAUM G
      EHT PLOETZLICH AUS{3SPACE}UND EIN ";      <000>
15023 PRINT"HELLER RECHTECKIGER FLECK ER-{
      3SPACE}SCHEINT AN DER ZIMMERWAND."        <215>
15024 PRINT"EIN VERSTECKTER PROJEKTOR BEGI
      NNT ZU{4SPACE}LAUFEN !"                   <068>
15025 PRINT"AN DER HAND ERSCHEINT DAS BILD
       EINES{4SPACE}WESENS DAS HALB MENSCH
       ";                                       <189>
15026 PRINT"UND HALB ROBOTER IST UND EIN T
      ONBAND BEGINNT ZU LAUFEN........"         <046>
15027 POKE 198,0:WAIT 198,1                     <011>
15030 PRINT"HIER SPRICHT PROFESSOR COHAN !
      {10SPACE}MEIN EXPERIMENT, EINE BES";      <192>
15031 PRINT"SERE KUENSTLI- CHE MENSCHHEIT
      ZU SCHAFEN IST FEHLGE-{3SPACE}SCHLAG
      EN."                                      <153>
15032 PRINT"MEINE ROBOTER HABEN SICH GEGEN
       MICH GE- WENDET."                        <072>
15033 PRINT"BEI EINEM KAMPF WURDE MEINE ME
      CHANISCHE KOERPERHAELFTE ZERSTOERT !
      "                                         <164>
15035 PRINT"ICH HABE NUR NOCH KURZE ZEIT Z
      U LEBEN."                                 <209>
15036 PRINT"ICH VERSUCHE NUN DIE VERBOTENE
       ZONE ZU{2SPACE}ERREICHEN UND DIE QU
      ELLE ";                                   <245>
15037 PRINT"DER MACHT MIT{2SPACE}EINER ZEI
      TBOMBE ZU ZERSTOEREN UM DIE BE-VORST
      EHENDE";                                  <217>
15038 PRINT" ROBOTERINVASION AUF DIE MEN-S
      CHHEIT ZU VERHINDERN."                    <082>
15040 PRINT"MOEGE GOTT MIR MEIN VERBRECHEN
       VERGEBEN......."                         <214>
15045 POKE 198,0:WAIT 198,1                     <029>
15050 PRINT"DAS BILD AN DER HAND VERSCHWIN
      DET UND{3SPACE}DAS LICHT GEHT WIEDER
      AN."                                      <186>
15099 RETURN                                    <169>
15100 DATA,,,40,,,,,,:PRINT"KLEINER WEISS
      GEKACHELTER RAUM."                        <050>
15120 IF VE<>45 AND VE<>35 OR OM<>36 THEN
      15130                                     <160>
15122 PRINT"DIE SCHLEUSSE OEFFNET SICH.":O
      $(28)="OFFENE SCHLEUSSE":TU(5)=0          <145>
15123 IF YB THEN YB=1:SC=SC+2.5                 <020>
15124 PRINT"EINE SYNTHETISCHE STIMME RUFT
      'ALARM !{2SPACE}SCHLEUSSE ZUR VERBOT
      ENEN";                                    <108>
15126 PRINT"{SHIFT-SPACE}ZONE IST AKTI- VI
      ERT WORDEN !'."                           <164>
15130 IF VE=13 AND OM=28 AND TU(5)=0 THEN
      RI(1)=52:RI=1                             <177>
15140 RETURN                                    <212>
15200 DATA,51,,,53,,,,,:PRINT"ENGER GANG I
      M INNEREN DES GEBIRGES."                  <211>
15205 PRINT"DER BODEN STRAHLT IN EINEM SEL
      TSAMEN{4SPACE}VIOLETTEN LICHT."           <188>
15220 IF GE(16)<>-1 THEN PA(9)=1                <102>
15230 RETURN                                    <046>
15300 DATA 54,,,,,,56,52,,:PRINT"HOEHLENSY
      STEM."                                    <090>
15320 RETURN                                    <136>
15400 DATA,53,,,,,55,,,:PRINT"HOEHLENSYSTE
      M."                                       <238>
15420 RETURN                                    <238>
15500 DATA 59,56,57,,,54,57,,,:PRINT"HOEHL
      ENSYSTEM."                                <092>
15520 RETURN                                    <082>
15600 DATA 55,58,57,,59,53,,,,:PRINT"HOEHL
      ENSYSTEM."                                <120>
15620 RETURN                                    <184>
15700 DATA 55,,,56,,55,,,,:PRINT"HOEHLENSY
      STEM."                                    <159>
15720 RETURN                                    <028>
15800 DATA 56,,,,57,,,,,:PRINT"HOEHLENSYST
      EM."                                      <092>
15820 RETURN                                    <128>
15900 DATA,,,55,,,,56,60,0:PRINT"HOEHLENSY
      STEM."                                    <251>
15920 RETURN                                    <230>
16000 DATA,,,,,,,,61,59:PRINT"HOEHLENSYSTE
      M."                                       <104>
16020 RETURN                                    <074>
16100 DATA,,,,,,,,,60:PRINT"AM ENDE DES WE
      GES.                                      <241>
16101 PRINT"VOR MIR LIEGT EIN TIEFER, CA 1
      0 METER{3SPACE}BREITER ABGRUND DER I
      N ";                                      <251>
16102 PRINT"DUNKELHEIT GE-{3SPACE}HUELLT I
      ST."                                      <198>
16103 PRINT"AUF DER ANDEREN SEITE DES ABGR
      UNDS SETZTSICH DER WEG FORT."             <211>
16104 PRINT"AN DER FELSWAND AUF DER ANDERE
      N SEITE{3SPACE}IST EIN ROT LEUCHTEND
      ER";                                      <118>
16105 PRINT" DRUCKKNOPF ANGE-BRACHT."           <080>
16120 IF VE=35 AND OM=39 THEN PRINT"MEINE
      ARME SIND NICHT 10 METER LANG !"          <009>
16130 IF BR<>1 THEN 16135                       <162>
16131 PRINT"EIN LAUTES RATTERN ERTOENT !":
      SC=SC+10                                  <132>
16132 PRINT"EINE BRUECKE AUS STAHL WIRD UE
      BER DEN{3SPACE}ABGRUND HINWEG HERRUE
      BER";                                     <107>
16133 PRINT" GESCHOBEN.":BR=2:OD(40)=61:OD
      (41)=63                                   <055>
16135 IF VE=13 AND OM=40 THEN RI(1)=62:RI=
      1                                         <221>
16140 IF VE<>8 OR GM<>6 OR OM<>39 THEN 161
      50                                        <220>
16141 IF BR<>-1 THEN PRINT"DIE PISTOLE FAE
      LLT HINAB.":GE(6)=0:GOTO 16150            <130>
16142 PRINT"OK.":PRINT"DIE PISTOLE TRIFFT
      DEN DRUCKKNOPF MIT{3SPACE}VOLER WUCH
      T !"                                      <159>
16143 GE(6)=63:BR=1:GOTO 16130                  <190>
16150 RETURN                                    <206>
16200 DATA 63,61,,,,,,,,:PRINT"AUF EINER S
      TAHLBRUECKE DIE UEBER EINEN ";            <150>
16210 PRINT" TIEFEN, DUNKLEN ABGRUND FUEHR
      T."                                       <145>
16220 RETURN                                    <020>
16300 DATA,,,64,0,0,0,0,0,0:PRINT"HOEHLENW
      EG AM RANDE EINES ABGRUNDS."              <117>
16320 IF VE=13 AND OM=41 THEN RI(1)=62:RI=
      1                                         <182>
16325 IF VE=35 AND OM=42 THEN PRINT"OK.":P
      RINT"ES PASSIERT NICHTS."                 <250>
16330 RETURN                                    <130>
16400 DATA,,63,,,,,65,,:PRINT"HOEHLENSYSTE
      M."                                       <087>
16420 RETURN                                    <222>
16500 DATA,,,,64,,,,66,:PRINT"FEUCHTE GROT
      TE."                                      <235>
16501 PRINT"EINE BREITE UNREGELMAESSIGE ST
      EINTREPPE FUEHRT NACH OBEN."              <019>
16520 RETURN                                    <066>
16600 DATA,,,67,,,,,,65:PRINT"RIESSIGER PR
      UNKVOLLER TEMPELRAUM."                    <073>
16620 RETURN                                    <166>
16700 DATA,69,66,68,,,,,,:PRINT"GROSSER RU
      NDER RAUM."                               <245>
16720 IF VE<>8 OR OM<>43 THEN 16740             <129>
16725 IF G1<>18 AND GE(G1)=-1 THEN PRINT"O
      K.":GE(G1)=0                              <023>
16727 IF G1<>18 THEN 16740                      <207>
16730 PRINT"EINE SYNTHETISCHE STIMME AUS D
      ER ZEIT-{2SPACE}BOMBE SPRICHT :":GE(
      18)=-9                                    <048>
16732 PRINT"'MISSION BEENDET. DETONATION E
      RFOLGT IN ";ZB;"ZEITEINHEITEN !'"         <182>
16733 SC=SC+10                                  <145>
16740 RETURN                                    <032>
16800 DATA,,67,,,,,,,:PRINT"ALTARRAUM."         <244>
16820 RETURN                                    <112>
16900 DATA 67,,,,,,,,,:PRINT"AUSSERHALB DE
      S TEMPELS."                               <062>
16920 IF VE=23 AND OM=44 THEN PRINT"DIE MA
      UER IST VIEL ZU HOCH UND ZU STEIL."       <195>
```

**Listing »Cohan's Land« (Fortsetzung)**

```
16925 IF VE=13 OR VE=26 OR VE=48 THEN RI(1
      )=75:RI=1:SC=SC+5                      <060>
16930 RETURN                                 <224>
17100 DATA,,,,,,,,,10:PRINT"HOCH OBEN AUF
      EINEM AFFENBROTBAUM."                  <118>
17110 IF PA(3)<>-1 THEN PA(3)=5              <176>
17120 RETURN                                 <158>
17200 DATA 39,45,,,,,,,,:PRINT"IN EINEM BE
      LUEFTUNGSSCHACHT."                     <011>
17220 RETURN                                 <004>
17500 DATA,,,,,,,,,:PRINT"ICH SCHWIMME UNT
      ER DER MAUER DURCH UND{2SPACE}TREIBE
       NUN";                                 <053>
17501 PRINT" AUF DEM FLUSS......"            <172>
17520 POKE 198,0:WAIT 198,1                  <220>
17525 :PRINT"{CLR}UND SO ENDETE DAS ABENTE
      UER DES{9SPACE}MR BOLINGS :"           <205>
17530 PRINT"MIT LETZTER KRAFT ERREICHTE ER
       DEN{6SPACE}INSELSTRAND, WO ER ";      <022>
17532 PRINT"VON EINEM VORBEIFAH- RENDEN HA
      NDELSSCHIFF ENTDECKT UND IN ";         <225>
17533 PRINT"{3SPACE}SEINE HEIMAT ZURUECKGE
      BRACHT WURDE."                         <225>
17540 IF GE(18)<>-9 THEN 17550               <181>
17545 PRINT"ALS DIE ZEITBOMBE EXPLODIERTE
      VERSANK{3SPACE}DIE GESAMTE INSEL SPU
      RLOS";                                 <087>
17546 PRINT" IM MEER......"                  <088>
17550 IF GE(18)=-9 THEN 17560                <225>
17552 PRINT"WAS MIT DEN ROBOTERN GESCHAH B
      LIEB{6SPACE}BISHER UNBEKANNT......     <142>
17560 PRINT"{WHITE}SIE HABEN";SC;"% DIESES
       ADVENTURES GELOEST."                  <237>
17564 PRINT"{2DOWN,LIG.RED}MIT FREUNDLICHE
      N GRUESSEN,                            <250>
17565 PRINT"{2DOWN,14SPACE}MICHAEL NICKLES   <010>
17566 POKE 198,0:WAIT 198,1:RUN              <177>
50000 REM ****************************       <206>
50050 IF UD=1 THEN 50220                     <139>
50055 TT=0                                   <050>
50060 FOR I=1 TO 10:BE$(I)="":NEXT:POKE 19
      8,0:BE$=""                             <090>
50061 PRINT"{GREEN}WAS SOLL ICH NUN MACHEN
       ?":PRINT"{GREY 2,DOWN}@";             <151>
50080 GET X$:TT=TT+1:IF TT>1500 THEN PRINT
       "{UP,GREEN}":GOTO 50140               <223>
50085 IF X$=""THEN 50080                     <040>
50086 TT=0                                   <081>
50090 IF PEEK(203)=1 OR LEN(BE$)>75 THEN I
      F BE$<>""THEN PRINT"{LEFT}.{GREEN}":
      GOTO 50140                             <187>
50100 I=ASC(X$):IF I<65 OR I>90 THEN IF I<
      >32 AND I<>20 AND I<>34 THEN 50080     <038>
50110 IF I=20 AND LEN(BE$)=0 THEN 50080      <093>
50120 IF I=20 THEN PRINT"{2LEFT,2SPACE,2LE
      FT}@";:BE$=LEFT$(BE$,LEN(BE$)-1):GOT
      O 50080                                <127>
50130 BE$=BE$+CHR$(I):I=I+128:PRINT"{LEFT}
      ";CHR$(I);"@";:GOTO 50080              <067>
50140 BE$=BE$+" ":FOR I=1 TO 10:BE$(I)="":
      NEXT:WZ=1:FOR I=1 TO LEN(BE$)          <131>
50150 :IF MID$(BE$,I,1)=" "THEN GOSUB 5019
      0:GOTO 50180                           <050>
50160 :IF WZ>19 THEN PRINT"EINGABE IST ZU
      LANG !":I=LEN(BE$)+1:GOTO 50180        <074>
50170 :BE$(WZ)=BE$(WZ)+MID$(BE$,I,1)         <127>
50180 NEXT I:GOTO 50220                      <112>
50190 IC=0:FOR I1=1 TO AZ:IF BE$(WZ)=AU$(I
      1)THEN IC=1                            <118>
50200 NEXT I1:IF IC=0 THEN WZ=WZ+1:RETURN    <189>
50210 BE$(WZ)="":RETURN                      <194>
50220 IF UD=1 THEN UD=0:GOTO 50240           <001>
50230 WZ=1:VE=0:OB=0:OM=0:PE=0               <231>
50240 IC=0:G1=0:G2=0:GM=0                    <164>
50241 B$=BE$(WZ):IF B$=""THEN RETURN         <171>
50242 IF B$="SUPERMIKE"THEN PRINT"09372106
      82":IC=1:GOTO 50370                    <137>
50245 FOR I=1 TO 10:IF B$=RI$(I)THEN RI=I    <126>
50246 NEXT I:IF RI<>0 THEN RETURN            <106>
50260 FOR I=1 TO VZ:IF B$=LEFT$(VE$(I),LEN
      (B$))THEN VE=I-VAL(RIGHT$(VE$(I),1))
      :IC=1                                  <001>
50261 NEXT I:IF IC THEN 50370                <028>
50270 FOR I=1 TO GZ                          <174>
50275 IF MID$(B$,2,LEN(B$)-1)<>MID$(GE$(I)
```

```
      ,2,LEN(GE$(I))-1)THEN 50300            <162>
50280 IC=1:IF G1=0 THEN G1=I:IF GE(G1)=-1
      THEN GM=I                              <210>
50290 G2=I:IF G2=G1 THEN G2=0                <002>
50300 NEXT I:IF IC THEN 50370                <067>
50310 FOR I=1 TO OZ:IF B$=OB$(I)THEN OB=I:
      IC=1:IF OO(I)=ZN THEN OM=I             <007>
50320 NEXT I:IF IC THEN 50370                <087>
50329 REM      IFLEN(B$)<3THEN50350          <017>
50330 FOR I=1 TO PZ                          <111>
50331 IF MID$(B$,2,LEN(B$)-1)=MID$(PE$(I),
      2,LEN(PE$(I))-1)THEN PE=I:IC=1         <144>
50332 IF PE=I AND PE(I)=ZN THEN PM=PE:IC=2   <113>
50340 NEXT I:IF IC THEN 50370                <107>
50350 IF B$="UND"THEN UD=1:IC=1              <126>
50360 IF IC=0 AND RI=0 THEN PRINT"ICH KENN
      E ";B$;" NICHT !":GOTO 50000           <187>
50370 WZ=WZ+1:IF WZ>10 OR BE$(WZ)=""OR UD=
      1 THEN RETURN                          <046>
50380 IC=0:GOTO 50241                        <040>
51000 REM ALLGEMEINE TABELLEN ----------     <148>
51005 RESTORE 51000                          <228>
51010 DATA N,S,W,O,NW,NO,SW,SO,RAUF,RUNTER
      :FOR I=1 TO 10:READ RI$(I):NEXT        <066>
51100 TU(1)=1:REM SCHRANK                    <067>
51105 TU(2)=1:REM HAUSTUER                   <159>
51110 TU(3)=1:REM TRUHE                      <195>
51115 TU(4)=2:REM SCHRANK                    <179>
51116 TU(5)=1:REM SCHLEUSSE                  <226>
51120 MU=4                                   <207>
51130 DIM BE$(20)                            <242>
51200 :                                      <121>
51210 DATA 51,51,51,40,41,42,44,45,48,48,4
      8,45,44,42,41,40,-1                    <167>
51220 DIM MA(17):FOR I=1 TO 17:READ MA(I):
      NEXT:MA=1:REM ROBOT AKTIV              <242>
52000 REM W O R T S C H A T Z ----------     <069>
52005 RESTORE 52000                          <228>
52010 DATA OEFFNE,SCHLIESSE,SCHAUE,UNTERSU
      CHE1,NIMM,NEHME1,HOLE2,VERLIERE,LEGE
      1                                      <124>
52011 DATA WIRF2,WERFE3,INVENTUR,GEHE        <058>
52012 DATA BETRETE1,VERLASSE,GIB,GEBE1,ZEI
      GE2,WARTE,TOETE,ERLEGE1,SCHIESSE2      <130>
52013 DATA KLETTERE,FAELLE,BAUE,SCHWIMME,U
      EBERQUERE,LIES,LESE1,HILFE             <251>
52014 DATA BEFESTIGE,BINDE1,VERKNOTE2,NAHO
      C,DRUECKE,BETAETIGE1                   <149>
52015 DATA BRECHE,BRICH1,ZERSCHLAGE2,ZERST
      OERE3,VERNICHTE4                       <110>
52016 DATA VERBINDE,KURZ1,STECKE,ZIEHE,SCH
      IEBE1,SETZE,SPRINGE,ESSE               <167>
52040 VZ=49:DIM VE$(VZ):FOR I=1 TO VZ:READ
       VE$(I):NEXT                           <156>
52100 DATA DER,DIE,DAS,DEN,IN,AUF,NACH,VOM
      ,ZUM,DEM,AUFS,SIE,MIT,UEBER,HOCH,WEG   <159>
52105 DATA EIN,AM,AN,DICH                    <001>
52120 AZ=20:DIM AU$(AZ):FOR I=1 TO AZ:READ
       AU$(I):NEXT                           <113>
52200 REM DIE OBJEKTE --------------         <195>
52210 DATA SCHRANK,"ALTER SCHRANK AUS EICH
      ENHOLZ",1                              <143>
52212 DATA HAUSTUER,"HAUSTUER",1             <219>
52214 DATA SCHIFF,"SCHIFF",3                 <048>
52216 DATA TRUHE,"GROSSE SEEMANNSTRUHE",0    <246>
52217 DATA TUER,"HAUSTUER",2                 <214>
52218 DATA BAUM,"HOHER AFFENBROTBAUM",10     <191>
52219 DATA MAUER,"SEHR HOHE, BREITE MAUER
      AUS FELSBROCKEN",8                     <218>
52220 DATA FLUSS,"BREITER REISSENDER FLUSS
      ",11                                   <106>
52221 DATA KROKODIL,"VIELE KROKODILE LAUER
      N IM FLUSS",11                         <081>
52222 DATA KAMIN,"KAMIN",1                   <011>
52223 DATA BAUM,"DUENNE HOLZBAUME",12        <077>
52224 DATA FLUSS,"BREITER REISSENDER FLUSS
      ",14                                   <136>
52225 DATA HOEHLE,"EINGANG ZU EINER HOEHLE
      ",25                                   <157>
52226 DATA HOEHLENWAND,"HOEHLENWAND",22      <137>
52227 DATA STEIN,"SPITZER KREIDESTEIN",0     <219>
52228 DATA SCHRIFT,"SCHWER LESBARE SCHRIFT
      ",0                                    <131>
52229 DATA SKELETT,"IN EINER ECKE LIEGT EI
      N MENSCHLICHES SKELETT",22             <125>
52230 DATA EISENRING,"AM BODEN IST EIN EIS
      ENRING BEFESTIGT",27                   <174>
```

```
52231 DATA SPALT,"CA 1 METER BREITER SPALT
       IN DER GEBIRGSWAND",31                <164>
52232 DATA TROPFSTEINE,"VON DER DECKE HAEN
      GEN GEBILDE AUS TROPFSTEINEN",35       <025>
52233 DATA STAHLTUER,"STAHLTUER",36          <104>
52234 DATA AUFZUG,"DIE STAHLTUER GIBT DEN
      ZUGANG ZU EINEM AUFZUG FREI",0         <141>
52235 DATA TASTE,"ROTE TASTE",37             <150>
52236 DATA SCHALTKASTEN,"AN DER WAND HAENG
      T EIN SCHALTKASTEN",37                 <084>
52237 DATA KABEL,"IM SCHALTKASTEN SIND ZWE
      I ROTE KABEL MIT BLANKEN ENDEN",0      <011>
52238 DATA STECKDOSE,"IN DER WAND IST EINE
       STECKDOSE",41                         <070>
52239 DATA SCHACHT,"IN DER WAND IST EIN BE
      LUEFTUNGSSCHACHT",39                   <171>
52240 DATA SCHLEUSSE,"SCHLEUSSE AUS STAHL"
      ,51                                    <006>
52241 DATA SCHRANK,"SCHRANK AUS STAHL",43    <096>
52242 DATA SCHACHT,"IN DER WAND IST EIN BE
      LUEFTUNGSCHACHT",45                    <134>
52244 DATA TREPPE,"WENDELTREPPE DIE NACH U
      NTEN FUEHRT",48                        <190>
52245 DATA TREPPE,"WENDELTREPPE DIE NACH O
      BEN FUEHRT",47                         <012>
52246 DATA STAHLTUER,"VERSCHLOSSENE STAHLT
      UER",38                                <220>
52247 DATA KAMPFROBOTER,"RIESSIGES REGUNGS
      LOSES KAMPFROBOTERHEER",47             <103>
52248 DATA DRAHTGITTER,"VOR DEM SCHACHT IS
      T EIN DRAHTGITTER",39                  <082>
52249 DATA SCHALTHEBEL,"NEBEN DER SCHLEUSS
      E IST EIN ROTER SCHALTHEBEL",51        <086>
52250 DATA TOTEN,"AM BODEN LIEGT EIN TOTER
       - HALB MENSCH HALB ROBOTER",51        <202>
52252 DATA SESSEL,"IN DER MITTE DES RAUMS
      STEHT EIN ANTIKER SESSEL",50           <161>
52253 DATA DRUCKKNOPF,"LEUCHTENDER DRUCKKN
      OPF",61                                <116>
52254 DATA BRUECKE,"BRUECKE AUS STAHL",0     <075>
52255 DATA BRUECKE,"BRUECKE AUS STAHL",0     <076>
52256 DATA DRUCKKNOPF,"LEUCHTENDER DRUCKKN
      OPF IN DER FELSWAND",63                <070>
52257 DATA BRUNNEN,"IN DER MITTE DES RAUMS
       IST EINE ART VON BRUNNEN",67          <239>
52258 DATA MAUER,"SEHR HOHE, BREITE MAUER
      AUS FELSBROCKEN",69                    <030>
52259 DATA FLUSS,"BREITER FLUSS, DER UNTER
       DER MAUER DURCHFLIESST",69            <056>
52260 OZ=45:DIM OB$(OZ):DIM O$(OZ):DIM OO(
      OZ)                                    <151>
52262 FOR I=1 TO OZ:READ OB$(I):READ O$(I)
      :READ OO(I):NEXT                       <226>
52300 DATA "SEEKARTE",0,"MUSCHELN",5         <083>
52302 DATA"SAEBEL",-3,"SCHIFFSZWIEBACK",-3
      ,"SEIL",-3,"PISTOLE",-3                <067>
52303 DATA"MUNITION",-3,"FLASCHE",-2,"BUCH
      ",0,"AXT",-3,"HOLZ",0                  <225>
52305 DATA"FLOSS",0,"BLECHDECKEL",0,"DRAHT
      ",0,"STROMKABEL",46                    <079>
52306 DATA"SCHUTZANZUG",-4,"SCHLUESSEL",0,
      "ZEITBOMBE",0                          <008>
52330 GZ=18:DIM GE$(GZ):DIM GE(GZ):FOR I=1
       TO GZ:READ GE$(I):READ GE(I):NEXT     <235>
52400 DATA"NAVIGATOR",4,"KAEPTEN GRAUBARD"
      ,4,"LOEWE",0,"VERBRANNTER ROBOTER",0 <030>
52410 PZ=4:DIM PE$(PZ):DIM PE(PZ):FOR I=1
      TO PZ:READ PE$(I):READ PE(I):NEXT      <094>
53000 RETURN                                 <225>
60000 REM MP                                 <226>
60004 FOR I=49152 TO 49206:READ A:POKE I,A
      :NEXT:SYS 49152:SYS 49191:SP=49152     <091>
60010 DATA 160,0,177,122,201,44,240,1,96,3
      2,115,0,32,49,192,165,20,32,23,192     <173>
60020 DATA 76,0,192,41,63,141,1,221,9,64,1
      41,1,221,173,1,221,48,251,96,169,127 <009>
60025 DATA 141,3,221,169,0,76,23,192,32,13
      8,173,76,247,183                       <180>
60045 RETURN:REM SYS49152,6,3,26,0,16,4,4,
      39,7,21,2,19,4                         <147>
61000 REM SPEAKEASY-SAETZE                   <200>
61010 REM SYSSP,19,37,4,57,23,3,63,7,4,23,
      44,41,55,13:RETURN                     <252>
61020 REM SYSSP,33,7,44,41,4,33,30,8,4,16,
      26,62,4,11,59,8,3,3,4:RETURN           <097>
61030 REM SYSSP,27,26,62,45,53,4,4,4,19,37
      ,4,27,6,55,55,7,4,4,16,19,55,17,52,4 <131>
61031 REM SYSSP,63,31,45,19,11,4:RETURN      <131>
```

**Listing »Cohan's Land« (Schluß)**

# Asterix und Obelix – die Odyssee

**Den Galliern ist ihr Zaubertrank ausgegangen. Versuchen Sie schnellstens, die Ingredenzen, die nötig sind, um ihn wieder zu brauen, aus dem fernen Afrika zu besorgen.**

Die Handlung dieses Adventures basiert auf dem Asterix-Heft »Die Odyssee«. Miraculix ist sein Vorrat an Steinöl ausgegangen, den er dringend benötigt, um seinen römerschlagenden Zaubertrank zu brauen. Der Spieler schlüpft nun in die Rolle von Asterix, der das Steinöl aus Mesopotamien holen muß. Zur Lösung des Spieles sind aber viele Gefahren und Rätsel zu bestehen. Starten Sie also als Asterix in dem kleinen gallischen Dorf, und suchen Sie den lebenswichtigen Zusatz des magischen Trankes.

Das Spiel beinhaltet 48 Räume, wobei aber nicht alle Räume verschiedene Bilder liefern. Aus diesem Grund kann man sich auch in der Wüste von Mesopotamien leicht verlaufen. Sie können ganze Sätze eingeben, aber auch Abkürzungen werden akzeptiert.

Richtungsangaben wie »NORD« oder »SÜDWEST« müssen mit den üblichen Abkürzungen wie »N« oder »SW« eingegeben werden.

Folgende Verben werden verstanden:

»NIMM, NEHME, WIRF, VERLIERE, WERFE, LEGE, BINDE, BEFESTIGE, VERKNOTE, KLOPFE, BERUEHRE, SCHLAGE, TOETE, VERPRUEGLE, WARTE, GIB, GEBE, REICHE, KAUFE, KLETTERE, GEHE, TRAGE, BENUTZE, TRINKE, SCHLUERFE, SPRINGE, SAGE, SCHAU, INVENTUR, LIST, SUCHE, ERKUNDE, LIES, OEFFNE, KITZEL, KITZLE, LOESE, ENTFERNE, FUELLE, SPEICHERE, SAVE, LADE, LOAD, RESTART«.

Sie werden erkennen, daß das Programm nach den Tips aus dem 64'er Sonderheft »Abenteuerspiele 1« entstanden ist, zum Beispiel die GOTO X-Routine und der Wortzerlegungsteil. ist, zum Beispiel die GOTO X-Routine und der Wortzerlegungsteil. Aus produktionstechnischen Gründen war es leider nicht mehr möglich, das Listing im Heft zu drucken. Das Programm ist aber auf der Programm-Service-Diskette zu diesem Heft erhältlich.

(Norbert Klein/dm)

# Inka – Schatzsuche am Amazonas

**In den Tiefen des Amazonas-Urwaldes wartet der legendäre Inka-Schatz darauf, um von Ihnen entdeckt und geborgen zu werden.**

Bei dem Programm Inka handelt es sich – von der Titelgrafik abgesehen – um ein reines Textadventure. Der Spieler schlüpft in die Rolle des Schatzsuchers Anthony Wolf, der von einem Museum beauftragt wird, den legendären Schatz der Inkas zu suchen. Außerdem soll er ein Zeichen eines verschollenen Forschers finden, der vor ihm sein Glück versuchte und seither als vermißt gilt. Sie fliegen mit einem Helikopter in das Zielgebiet und springen mit einem Fallschirm mitten in den Urwald. Hier beginnt das eigentliche Adventure. Inka verfügt über mehr als 50 Räume mit fast 40 Rätseln. Das Vokabular beläuft sich auf etwa 100 Verben und Nomen. Wegen des großen Sprachschatzes wurde auf eine Wortliste verzichtet. Sämtliche Befehle können abgekürzt werden (»GE« für »GEHE« oder »U« für »UHR«). Der Klammeraffe (@) bewirkt eine Wiedergabe des zuletzt eingegebenen Befehls.

Der Spielstand kann durch »SAVE« oder »LOAD« auf Kassette oder Diskette gespeichert und geladen werden.

(T. Mahnig/W. Borucki/dm)

```
Du stehst im undurchsichtigen Gestruepp
des lianenbewachsenen Dschungels.
Herumliegende Gegenstaende:
-fallschirm

Ausgaenge:s nw no

Was nun ? nw<

Du stehst im undurchsichtigen Gestruepp
des lianenbewachsenen Dschungels.
Herumliegende Gegenstaende:
keine

Ausgaenge:w o sw so

Was nun ? <
```

**Listing »Inka«**

```
10 REM"{7SPACE}***INKA***                      <082>
20 REM"{6SPACE}(C) BY BCSPC                    <200>
30 REM"{10SPACE}1985                           <079>
31 REM"DESIGNED BY PILE & ZOMBIE               <117>
32 REM"{2SPACE}PROGRAMMED{2SPACE}BY ZOMBIE     <078>
33 REM"                                        <112>
34 REM"{6SPACE}THILO MAHNIG                    <078>
35 REM"{2SPACE}A.D. STEPHANSBERG 56            <213>
36 REM"{4SPACE}5309 MECKENHEIM                 <070>
37 REM"{3SPACE}TEL.:{2SPACE}02225/4721         <147>
38 REM"------------------------               <059>
45 :                                           <021>
46 :                                           <022>
47 :                                           <023>
48 :                                           <024>
49 RESTORE:R=RND(-TI)                          <114>
50 DATA 76,24,1,177,251,145,251,200,208,24
   9,230,252,202,208,244,96,120,160,0,169      <058>
52 DATA 160,132,251,133,252,162,32,32,11,1
   ,169,224,132,251,133,252,162,32,32,11       <097>
54 DATA 1,169,53,133,1,88,96:FOR I=264 TO
   310:READ X:POKE I,X:NEXT:SYS 264            <005>
56 FOR I=710 TO 730:READ X:POKE I,X:NEXT       <176>
58 DATA 208,3,76,29,168,32,192,2,32,19,166
   ,56,165,95,233,1,164,96,76,36,168           <184>
60 POKE 40996,197:POKE 40997,2:POKE 1,54       <232>
62 FOR I=43168 TO 43170:READ X:POKE I,X:NE
   XT                                          <077>
64 FOR I=704 TO 709:READ X:POKE I,X:NEXT       <182>
66 DATA 32,192,2,32,138,173,76,247,183         <045>
70 :                                           <046>
71 :                                           <047>
72 :                                           <048>
100 POKE 53280,7:POKE 53281,7                  <014>
110 PRINT"{CLR,BLACK,CTRL-H}"CHR$(142):REM
     H:CTRL+H                                  <099>
120 PRINT                                      <222>
130 GOTO 2000                                  <078>
499 REM INPUT                                  <234>
500 PRINT:GOSUB 700                            <015>
510 Q=0:FOR I=1 TO LEN(II$):IF MID$(II$,I,
    1)=" "THEN Q=I:I=LEN(II$)                  <235>
520 NEXT:IF Q=0 THEN V$=II$:N$="":GOTO 540 <082>
530 V$=LEFT$(II$,Q-1):N$=MID$(II$,Q+1)         <236>
532 Q1=0:FOR I=1 TO LEN(N$):IF MID$(N$,I,1
    )=" "THEN Q1=I                             <172>
534 NEXT:IF Q1=0 THEN O$="":O=0:GOTO 540       <021>
536 O$=MID$(N$,Q1+1):N$=LEFT$(N$,Q1-1)         <060>
540 L=LEN(V$):V=0:FOR I=1 TO 54:IF V$=LEFT
    $(V$(I),L)THEN V=I:I=54                    <034>
550 NEXT:IF V=0 THEN PRINT"ICH VERSTEHE DE
    IN VERB NICHT!":GOTO 500                   <239>
560 IF RIGHT$(V$(V),1)="*"THEN V=V-1:GOTO
    560                                        <109>
570 IF Q=0 THEN N=0:RETURN                     <165>
580 L=LEN(N$):N=0:FOR I=1 TO MX:IF N$=LEFT
    $(N$(I),L)THEN N=I:I=MX                    <168>
590 NEXT:IF N=0 THEN PRINT"ICH VERSTEHE DE
    IN NOMEN NICHT!":GOTO 500                  <071>
600 IF O$=""THEN RETURN                        <249>
610 L=LEN(O$):O=0:FOR I=1 TO MX:IF O$=LEFT
    $(N$(I),L)THEN O=I:I=MX                    <170>
620 NEXT:IF O=0 THEN PRINT"ICH VERSTEHE DE
    IN OBJEKT NICHT!":GOTO 500                 <021>
630 RETURN                                     <180>
700 PRINT:PRINT"WAS NUN ? ";:I1$=II$:II$="
    "                                          <090>
710 PRINT"<{LEFT}";                            <220>
720 GET IN$:IF IN$=""THEN 710                  <228>
730 IF IN$=CHR$(20)AND II$>""THEN 770          <136>
740 IF IN$=CHR$(13)THEN 780                    <124>
750 IF(IN$<"@"OR IN$>"Z")AND IN$<>" "OR LE
    N(II$)>27 THEN 720                         <189>
755 IF IN$="@"THEN 795                         <168>
760 II$=II$+IN$:PRINT IN$;:GOTO 710            <209>
770 II$=LEFT$(II$,LEN(II$)-1):PRINT IN$;:G
    OTO 710                                    <029>
780 :                                          <250>
785 IF II$=""THEN 720                          <069>
790 PRINT:RETURN                               <059>
795 II$=I1$:PRINT CHR$(141)"{UP,10RIGHT}"I
    I$;:GOTO 710                               <176>
999 REM SIEH                                   <019>
1000 PRINT"HERUMLIEGENDE GEGENSTAENDE:":F=
     0                                         <235>
1010 FOR I=1 TO 23:IF N(I)=P AND NF(I)<>P
     THEN F=1:PRINT"-";N$(I)                   <064>
1020 NEXT:IF F=0 THEN PRINT"KEINE"             <177>
1030 IF N(46)=P AND P<>14 THEN PRINT"DER A
     FFE IST HIER."                            <000>
1050 PRINT:PRINT"AUSGAENGE:";:RESTORE 4900
     +P*100:X=.                                <098>
1060 READ X,E:PRINT V$(ABS(X))" ";:E(ABS(X
     ))=E:IF X>0 THEN 1060                     <004>
1065 IF X=.THEN PRINT"KEINE";                  <142>
1070 PRINT:RETURN                              <085>
1500 PRINT"{3DOWN}"                            <229>
```

```
1510 PRINT TAB(17)"{RVSON,RIGHT}+"                          <208>
1520 PRINT TAB(17)"{RVSON}RIP"                              <087>
1530 PRINT TAB(17)"{RVSON,RIGHT,SPACE}"                     <175>
1540 PRINT TAB(17)"{RVSON,RIGHT,SPACE}"                     <187>
1550 PRINT"{3DOWN}WILLST DU NOCHMAL SPIELE
     N ?"                                                   <098>
1560 GET IN$:IF IN$=""THEN 1560                             <001>
1570 IF IN$="J"THEN PRINT"{CLR}":RUN 3000                   <048>
1580 IF IN$<>"N"THEN 1560                                   <189>
1590 PRINT"{CLR}":END                                       <004>
1999 END                                                    <223>
2000 PRINT"{CLR}";:POKE 53265,PEEK(53265)A
     ND 239                                                 <037>
2002 PRINT"BLACK CROCO SOFTWARE PROGRAMMIN
     G CREW{5SPACE}PRESENTS:{SPACE,DOWN,4S
     PACE}INKA"                                             <177>
2003 FOR I=54272 TO 54296:POKE I,0:NEXT                     <201>
2005 PRINT"{3SPACE}PPPP OL P PPPPP                          <095>
2010 PRINT"{3SPACE}L P MGNNNNP PPN                          <158>
2020 PRINT"{3SPACE}NNNNM{3SPACE}GGGGG                       <080>
2030 PRINT"{3SPACE}NNGN GNMMGY G                            <205>
2040 PRINT"NYYY{2SPACE}N GN M OGG                           <077>
2050 PRINT"YYYYYPG YY{2SPACE}Y Y                            <031>
2055 REM ++++++++-++++++++++-+++++++++-                     <230>
2060 PRINT"{3DOWN,4SPACE}UI":PRINT"{4SPACE
     }JK"                                                   <051>
2065 RESTORE 30000:FOR I=0 TO 62:READ X:PO
     KE 832+I,X:NEXT:V=53248:POKE V+39,0                    <171>
2066 POKE V+40,0:POKE V+41,0:POKE V+2,100:
     POKE V+3,80:POKE 2041,15:POKE 2042,14                  <242>
2067 POKE V+16,2                                            <216>
2068 FOR I=0 TO 62:READ X:POKE 896+I,X:NEX
     T:FOR I=0 TO 62:READ X:POKE 960+I,X:N
     EXT                                                    <239>
2070 POKE 2040,13:POKE V,200:POKE V+1,65:P
     OKE V+29,3:POKE V+21,3:POKE V+23,2                     <004>
2080 SI=54272:POKE SI+24,15:POKE SI+5,0:PO
     KE SI+6,240:POKE SI+1,10                               <165>
2100 PRINT TAB(10)"{7SPACE}PNMP                             <059>
2110 PRINT TAB(10)"{6SPACE}NN NNG                           <187>
2120 PRINT TAB(10)"{5SPACE}NN NNNMP                         <251>
2130 PRINT"PPPPPPPPPPPPPPNNMNNN NNLPPPPPPP
     PPPPPPPP"                                              <110>
2140 PRINT TAB(10)"{3SPACE}NOZZPN NNNM                      <122>
2150 PRINT TAB(10)"{2SPACE}NN{4SPACE}MNNN
     N                                                      <253>
2160 PRINT TAB(10)"{2SPACE}O{2SPACE}OP{2SP
     ACE}PN N                                               <115>
2170 PRINT TAB(10)" N{2SPACE}ODFP{2SPACE}M
     N                                                      <246>
2180 PRINT TAB(10)" TTTTTTTTTT                              <113>
2190 POKE 53265,PEEK(53265)OR 16                            <222>
2200 Z=0:FOR I=90 TO 0 STEP-1:POKE V+2,I:P
     OKE 975+Z,255:POKE 976-Z,0:Z=1-Z                       <099>
2210 POKE SI+4,128+Z:FOR N=1 TO 20:NEXT N,
     I                                                      <249>
2220 POKE V+4,255:POKE V+5,100:POKE V+21,7                  <013>
2230 FOR I=100 TO 200:POKE 975+Z,255:POKE
     976-Z,0:Z=1-Z:POKE V+5,I:POKE SI+4,12
     8+Z                                                    <100>
2240 FOR N=1 TO 20:NEXT N,I                                 <230>
2250 FOR I=80 TO 0 STEP-1:POKE 975+Z,255:P
     OKE 976-Z,0:POKE SI+4,128+Z:Z=1-Z                      <185>
2260 POKE V+3,I:IF I<32 THEN POKE SI+24,I/
     2                                                      <053>
2265 FOR N=1 TO 20:NEXT N,I                                 <255>
2270 POKE SI+4,0:POKE V+21,0                                <040>
2280 PRINT"{CLR,CTRL-N,9DOWN}VON HENZEL BO
     RUCKI & THILO HAHNIG"                                  <159>
2290 POKE 198,0:WAIT 198,1                                  <230>
2300 PRINT"{CLR,CTRL-N,DOWN}DAS SPIEL:":RE
     M N:CTRL+N                                             <177>
2310 PRINT"YYYYYYYYYY"                                      <130>
2320 PRINT"{DOWN}DU BIST ANTHONY WOLF.                      <029>
2330 PRINT"{DOWN}DU WURDEST MIT DEM FALLSC
     HIRM UEBER DEM"                                        <035>
2340 PRINT"REGENWALD DES AMAZONAS ABGEWORF
     EN."                                                   <209>
2350 PRINT"{DOWN}FINDE{SHIFT-SPACE}DEN{SHI
     FT-SPACE}SCHATZ{SHIFT-SPACE}DER{SHIFT
     -SPACE}INKA{SHIFT-SPACE}!"                             <009>
2360 PRINT"{DOWN}BRINGE EIN ZEICHEN DES FO
     RSCHERS, DER"                                          <255>
2370 PRINT"SEINE STATION IRGENDWO DORT UNT
     EN HAT!"                                               <105>
2380 PRINT"{DOWN}HAST DU BEIDE AUFGABEN ER
     FUELLT, SO"                                            <227>
2390 PRINT"MACHE EIN WEITHIN SICHTBARES SI
     GNAL, UND";                                            <107>
2400 PRINT"DER HUBSCHRAUBER WIRD DICH WIED
     ER ABHO- LEN!"                                         <211>
2410 PRINT"{DOWN}DRUECKE EINE TASTE."                       <148>
2420 POKE 198,0:WAIT 198,1:GET A$                           <058>
3000 FL=22:DIM FL%(FL):RESTORE 3020                         <137>
3005 PRINT"{CLR,CTRL-N}"                                    <204>
3010 DIM V$(54):FOR I=1 TO 54:READ V$(I):N
     EXT                                                    <055>
3020 DATA N,S,O,W,NO,NW,SO,SW,NIMM,GIB,LEG
     E,SCHWIMME,STUELPE,WIRF,SPRINGE                        <177>
3030 DATA SCHLAGE,TOETE*,SCHAU,SIEH*,BETRA
     CHTE*,GUCKE*,HALTE,INVENTUR,LISTE*                     <197>
3040 DATA ZIEHE,DRUECKE*,GEHE,STELLE,KRIEC
     HE,KRABBLE*,GRABE,BUDDLE*,SCHLIESSE                    <076>
3050 DATA OEFFNE,BINDE,BEFESTIGE*,KLETTERE
     ,HANGLE,WASCHE,SCHMIERE,KLEBE*                         <246>
3060 DATA BESTREICHE*,BESCHMIERE*,SAVE,LOA
     D,QUIT,ENDE*,STOP*,MACHE,ENTZUENDE*                    <238>
3070 DATA WARTE,TRAGE,*ANZIEHE,JA                           <116>
3100 MX=51:DIM N$(MX),N(MX),NF(MX):FOR I=1
      TO MX:READ N$(I),N(I):NF(I)=N(I):NEX
     T                                                      <045>
3110 DATA MACHETE,0,UHR,43,STEIN,-1,EIMER,
     5,BANANEN,46,BRILLE,38,TOPF,8                          <052>
3120 DATA LIANE,-1,TASCHENTUCH,38,EIER,42,
     GOLD,-1,SCHLUESSEL,39,ALLES,-1                         <008>
3130 DATA DECKEL,-1,TIERKADAVER,39,FIGUR,5
     0,SCHATZ,-1,SCHAUFEL,26                                <236>
3140 DATA FALLSCHIRM,0,STANGE,41,TRUHE,-1,
     GELD,0,TELLER,30                                       <119>
3200 DATA BAUM,,BIENEN,11,INKA,,TOTER,,HEB
     EL,,MESTIZE,,WASSERFALL,                               <219>
3210 DATA FLUSS,,STEINE,,INDIANER,,FEUER,                   <152>
3220 DATA SKORPION,,HONIG,,EINSIEDLER,,NES
     T,,AST,,TEMPELPORTAL,,SPALT,                           <024>
3230 DATA BRUNNEN,,HOEHLE,,FELS,,GEBUESCH,
     ,AFFE,14,NISCHE,,BRUECKE,                              <064>
3240 DATA ZELT,,STAUDE,,SEIL,                               <147>
3300 RESTORE 3310:FOR I=1 TO INT(RND(1)*8)
     +1:READ P:NEXT:N(19)=P                                 <101>
3310 DATA 26,29,7,3,12,24,17,40                             <018>
3320 KN$="KANN ICH NICHT.":WD$="WAS DENN ?
     ":HN$="HAB ICH NICHT."                                 <234>
3330 SN$="SEH ICH NICHT.":OK$="OK."                         <022>
4000 IF FL%(18)=.OR P=P1 THEN 4050                          <000>
4010 PRINT"DER AFFE KOMMT DIR NACH.":P1=P:
     N(46)=P:FL%(18)=.                                      <067>
4050 FOR I=1 TO 8:E(I)=.:NEXT:PRINT:GOTO 4
     900+P*100:PRINT                                        <203>
4100 GOSUB 1000                                             <248>
4110 GOSUB 4300:IF D THEN 1500                              <006>
4120 GOSUB 500:GOTO 19900+100*V                             <024>
4300 D=0                                                    <149>
4310 IF NOT(P>9 AND P<23 AND P<>18 AND N(1
     2)=0 AND FL%(3)=0)THEN 4322                            <037>
4320 PRINT:PRINT"EINE DIEBISCHE ELSTER STI
     EHLT DIR DEINENSCHLUESSEL.":N(12)=43                   <192>
4322 IF NOT(P>9 AND P<23 AND P<>18 AND N(2
     )=0 AND FL%(3)=0)THEN 4330                             <126>
4325 PRINT:PRINT"DIE ELSTER STIEHLT DIR DI
     E UHR.":N(2)=43                                        <126>
4330 IF FL%(2)=.THEN 4340                                   <070>
4332 FL%(2)=FL%(2)-1                                        <174>
4335 IF FL%(2)=.AND P=9 AND N(3)THEN PRINT
     "ER SETZT SICH WIEDER HIN.":N(3)=-1                    <128>
4340 IF FL%(4)=.THEN 4350                                   <145>
4342 FL%(4)=FL%(4)-1:IF FL%(4)>1 THEN PRIN
     T"DIR BLEIBT NICHT VIEL ZEIT...":GOTO
     4350                                                   <117>
4343 IF N(19)>-1 THEN 4348                                  <028>
4344 PRINT"DU WIRST HINUNTERGEWORFEN.DURCH
     DEN"                                                   <027>
4345 PRINT"FALLSCHIRM WIRST DU GERETTET:DI
     E INDIA-"                                              <254>
4346 PRINT"NER KOENNEN NUR FASSUNGSLOS STA
     RREN...":P=41:FL%(4)=0:FL%(22)=1:RETU
     RN                                                     <039>
4348 PRINT"DIE INDIANER WERFEN DICH IN DEN
     ABGRUND.":D=1                                          <072>
4350 IF(P<10 OR P>23)AND P<>43 THEN FL%(15
     )=0:GOTO 4380                                          <009>
4355 IF FL%(15)=-1 THEN 4380                                <130>
4360 FL%(15)=FL%(15)-1:IF FL%(15)=1 THEN P
     RINT"DU HOERST EIN GERAEUSCH..."                       <090>
4370 IF FL%(15)THEN 4380                                    <170>
```

**Listing »Inka« (Fortsetzung)**

```
4375 PRINT"EIN INDIANER STUERZT HERBEI UND
     UEBER-{2SPACE}WAELTIGT DICH!":D=1        <010>
4380 IF FL%(18)=.OR FL%(19)=.THEN 4400        <227>
4385 FL%(18)=FL%(18)-1                        <137>
4400 IF P<>42 THEN 4420                       <012>
4405 IF N(46)<>P AND N(46)<>-1 THEN PRINT"
     DER ADLER TOETET DICH.":D=1:RETURN      <192>
4408 IF FL%(20)THEN 4420                      <155>
4410 PRINT"DER ADLER TOETET DEN AFFEN.":FL
     %(20)=1:N(46)=-1                         <081>
4420 IF FL%(19)=.THEN N(5)=46                 <140>
4430 IF FL%(21)THEN FL%(21)=FL%(21)-1         <253>
4435 RETURN                                   <175>
5000 DATA -1,3                                <137>
5010 PRINT"DU STEHST AM FLUSSUFER, NAHE BE
     I EINEM{2SPACE}TOSENDEN WASSERFALL."    <122>
5020 PRINT"EIN SCHMALER UFERPFAD FUEHRT NO
     RDWAERTS.";:GOTO 4100                    <001>
5100 DATA -4,1                                <157>
5110 PRINT"DU BIST IN EINER HOEHLE HINTER
     DEM WAS- SERFALL."                       <017>
5120 IF FL%(0)=0 THEN PRINT"EIN INDIANER S
     TARRT DICH BOESARTIG AN!"                <086>
5130 IF FL%(0)THEN PRINT"ZWISCHEN DEN FELS
     EN IST EIN HEBEL."                       <019>
5140 GOTO 4100                                <040>
5200 DATA 2,1,-5,4                            <253>
5210 PRINT"DU STEHST AM FLUSSUFER. DER UFE
     RPFAD"                                   <229>
5220 PRINT"WEIST DEN WEG NACH NORDOSTEN UN
     D SUEDEN.";:GOTO 4100                    <032>
5300 DATA 8,3,-6,5                            <214>
5310 PRINT"DU BIST AM FLUSSUFER. DER FLUSS
      HAT"                                    <132>
5320 PRINT"EINIGE STEINE HERANGESPUELT.":G
     OTO 4100                                 <211>
5400 DATA 7,4,-1,6                            <083>
5410 PRINT"VOR DIR STEHT EIN BRUNNEN. EIN
     BREIT"                                   <122>
5420 PRINT"GRINSENDER MESTIZE HAELT DIR EI
     N SCHILD VOR DIE NASE:"                  <066>
5430 PRINT">FUER ABENTEURER $10.-<":GOTO 4
     100                                      <187>
5500 DATA 2,5,-1,7                            <179>
5510 PRINT"DU BIST IM TROCKENEN UNTERHOLZ.
     ":GOTO 4100                              <079>
5600 DATA 4,8,1,9,-2,6                        <017>
5610 PRINT"DEIN WEG HAT DICH IN DIE BERGE
     GEFUEHRT.";                              <207>
5620 PRINT"WESTLICH IST EIN HOEHLENEINGANG
     .":GOTO 4100                             <162>
5700 DATA -3,7                                <108>
5710 PRINT"DU HAST EINE ENGE HOEHLE BETRET
     EN."                                     <238>
5720 IF FL%(1)THEN PRINT"DER EIMER IST UEB
     ER DEM SKORPION.":GOTO 5750              <139>
5730 PRINT"VOR DEINEN FUESSEN HOCKT EIN ET
     WA FAUST-";                              <238>
5740 PRINT"GROSSER SKORPION,DER IM HALBDUN
     KEL DER{2SPACE}HOEHLE BLAEULICH SCHIM
     MERT."                                   <129>
5750 PRINT"IN EINER NISCHE ENTDECKST DU EI
     NEN TOPF HONIG."                         <060>
5760 GOTO 4100                                <152>
5800 DATA 2,7,-1,12                           <154>
5810 PRINT"HIER IST EIN EINSIEDLER."          <092>
5820 IF N(3)=-1 THEN PRINT"ER SITZT AUF EI
     NEM STEIN."                              <224>
5830 GOTO 4100                                <222>
5900 DATA -3,11                               <077>
5910 PRINT"DU STEHST IM UNDURCHSICHTIGEN G
     ESTRUEPP"                                <206>
5920 PRINT"DES LIANENBEWACHSENEN DSCHUNGEL
     S.":GOTO 4100                            <122>
6000 DATA 4,10,3,12,-1,15                     <155>
6010 IF N(25)<>11 THEN 5910                   <138>
6020 PRINT"EIN SURRENDER BIENENSCHWARM UMS
     CHWAERMT DICH."                          <080>
6030 IF N(36)=.THEN PRINT"DIE BIENEN STUER
     ZEN SICH AUF DEN HONIG."                 <160>
6040 GOTO 4100                                <178>
6100 DATA 2,9,4,11,-3,13                      <112>
6110 GOTO 5910                                <160>
6200 DATA 1,17,5,18,6,16,-4,12                <094>
6210 GOTO 5910                                <006>
6300 DATA 2,10,-5,20                          <080>
6310 IF N(46)<>14 THEN 5910                   <080>
6320 PRINT"EIN AFFE GAFFT DICH HUNGRIG AN.
     ..":GOTO 4100                            <104>
6400 DATA 2,11,-5,21                          <006>
6410 GOTO 5910                                <208>
6500 DATA 1,21,-7,13                          <158>
6503 IF FL%(3)THEN 6520                       <038>
6505 PRINT"EINE ELSTER SITZT AUF EINEM AST
      EINES"                                  <168>
6508 PRINT"BAUMES.ALS SIE DICH SIEHT, FLIE
     GT SIE{3SPACE}DAVON."                    <127>
6510 GOTO 5910                                <052>
6520 PRINT"DIE ELSTER KLEBT MIT IHREN KRAL
     LEN AM{3SPACE}AST FEST.":GOTO 5910       <150>
6600 DATA 2,13,6,21,-5,23                     <164>
6610 GOTO 5910                                <152>
6700 DATA 8,13,1,23,-5,24                     <071>
6710 PRINT"DU HAST EIN KLEINES DORF MIT IN
     KAHUETTENBETRETEN.":GOTO 4100            <220>
6800 DATA 2,14,-3,20                          <069>
6810 GOTO 5910                                <098>
6900 DATA 4,19,8,14,-3,21                     <245>
6910 GOTO 5910                                <198>
7000 DATA 4,20,3,22,8,15,-7,17                <102>
7010 GOTO 5910                                <044>
7100 DATA 4,21,-3,23                          <018>
7110 GOTO 5910                                <144>
7200 DATA 2,18,8,17,-3,24                     <232>
7210 GOTO 5910                                <246>
7300 DATA 8,18,4,23,-7,25                     <106>
7310 PRINT"DU STEHST AUF EINEM SEICHT BEWA
     LDETEN"                                  <165>
7320 PRINT"HUEGEL.WESTLICH IST DSCHUNGEL,S
     UEDWEST-"                                <047>
7330 PRINT"LICH ERBLICKST DU EIN INKADORF,
     SUEDOEST-";                              <178>
7340 PRINT"LICH IST EINE WACKELIGE HAENGEB
     RUECKE.":GOTO 4100                       <096>
7400 DATA 6,24,-7,26                          <073>
7410 PRINT"DU STEHST AUF EINER SCHWANKENDE
     N HAENGE-";                              <204>
7415 PRINT"BRUECKE. UNTER DIR ZUENGELN DIE
      ZISCHEN-";                              <240>
7420 PRINT"DEN FLUTEN DES FLUSSES DAHIN, D
     ER FUER"                                 <105>
7430 PRINT"SEINEN GOLDGEHALT BERUEHMT IST.
     ":GOTO 4100                              <069>
7500 DATA 6,25,1,27,-2,29                     <103>
7510 PRINT"DU BIST AM UFER DES FLUSSES.NOR
     DWESTLICH";                              <129>
7520 PRINT"IST EINE BRUECKE.";:IF N(18)<>2
     6 THEN PRINT:GOTO 4100                   <158>
7530 PRINT"IN DEN SAND IST NACH-","LAESSIG
      EINE SCHAUFEL GESTECKT.":GOTO 4100      <249>
7600 DATA -2,26                               <176>
7610 PRINT"HIER, OBERHALB DER BRUECKE IST
     DAS WAS- SER SEICHT UND KLAR.":GOTO 4
     100                                      <155>
7700 DATA 0,0                                 <167>
7710 PRINT"DU STEHST BIS ZUR HUEFTE IM WAS
     SER.":GOTO 4100                          <008>
7800 DATA 1,26,8,31,-4,30                     <248>
7810 PRINT"DU BIST AUF EINEM PFAD VOR EINE
     M ZELT."                                 <063>
7820 PRINT"DAS MUSS DIE BEHAUSUNG DES FORS
     CHERS{4SPACE}SEIN!":GOTO 4100            <228>
7900 DATA -3,29                               <111>
7910 PRINT"DU BIST IN DEM ZELT DES FORSCHE
     RS.ES IST";                              <184>
7920 PRINT"NIEMAND HIER.";:IF N(23)<>30 TH
     EN PRINT:GOTO 4100                       <074>
7930 PRINT"EIN TELLER MIT ESSEN STEHT AUF
     DEM BODEN.":GOTO 4100                    <060>
8000 DATA -5,29                               <245>
8010 PRINT"DU STEHST AUF EINER FELSIGEN HO
     EHE VOR"                                 <038>
8020 PRINT"EINEM ABGRUND.TIEF UNTEN SIEHST
      DU DEN"                                 <103>
8030 PRINT"TEMPEL.IN DER FELSWAND IST EIN
     ADLER-{3SPACE}NEST.":GOTO 4100           <087>
8100 DATA -7,34                               <055>
8110 PRINT"DU STEHST AN EINER TIEFEN SCHLU
     CHT,AN"                                  <010>
8120 PRINT"DEREN BODEN DU DEN INKATEMPEL Z
     U ERKEN-"                                <138>
8130 PRINT"NEN GLAUBST.":GOTO 4100            <120>
```

```
8200 DATA 6,29,-2,34                                  <042>
8210 PRINT"EIN FAST UNSICHTBARER PFAD WIND
     ET SICH"                                         <174>
8220 PRINT"DURCH DAS GESTRUEPP.ICH GLAUBE
     NICHT,"                                          <194>
8230 PRINT"DASS ICH IHN WIEDERFINDEN WUERD
     E.ER "                                           <207>
8240 PRINT"SETZT SICH NACH NORDEN UND SUED
     EN FORT.":GOTO 4100                              <190>
8300 DATA 1,33,-6,32                                  <189>
8310 PRINT"DU STEHST VOR EINEM TIEFEN ABGR
     UND."                                            <025>
8320 PRINT"UNTEN LIEGEN SKELETTE VON OPFER
     N."                                              <057>
8325 IF FL%(4)=.THEN PRINT"BEWAFFNETE INKA
     S STEHEN UM DICH HERUM."                         <006>
8340 GOTO 4100                                        <009>
8400 DATA -4,36                                       <225>
8410 PRINT"DU BIST AM FUSS DES ABGRUNDS.WE
     STLICH{3SPACE}IST DER TEMPEL.":GOTO 4
     100                                              <095>
8500 DATA -3,35                                       <011>
8510 PRINT"VOR DIR IST DAS PORTAL DAS TEMP
     ELS."                                            <114>
8520 PRINT"DAVOR LIEGT DER TOTE FORSCHER,A
     UF SEINEM";                                      <218>
8530 PRINT"GESICHT EIN SCHMERZVERZERRTER A
     USDRUCK.":IF FL%(14)=.THEN 4110                  <114>
8540 PRINT"DER TEMPELPORTAL IST OFFEN.":GO
     TO 4110                                          <211>
8600 DATA -6,4                                        <070>
8610 PRINT"DU STEHST AUF EINEM STEIN IM FL
     USS."                                            <204>
8620 IF FL%(5)=.THEN PRINT"DER NAECHSTE ST
     EIN IST EIN GUTES STUECK ENTFERNT."              <136>
8630 GOTO 4100                                        <228>
8700 DATA 5,40,-3,39                                  <206>
8710 PRINT"DU BIST AM OESTLICHEN FLUSSUFER
     ."                                               <219>
8720 IF N(6)<>38 AND N(9)<>38 THEN 4100               <213>
8730 IF N(6)=38 THEN PRINT"EINE ZERSCHLAGE
     NE BRILLE ":IF N(9)=38 THEN PRINT"UND
      EIN"                                            <170>
8740 IF N(6)<>38 THEN PRINT"EIN ";                    <023>
8750 IF N(9)=38 THEN PRINT"BLUTIGES TASCHE
     NTUCH MIT INITIALEN"                             <138>
8760 PRINT"LIEGT AUF DEM BODEN.":                     <227>
8770 GOTO 4100                                        <114>
8800 DATA 0,0                                         <251>
8810 PRINT"DU BIST IN EINE FALLGRUBE GERAT
     EN!"                                             <151>
8820 PRINT"MIT DIR HIER IST EIN TIERKADAVE
     R.";                                             <249>
8825 IF N(12)=39 THEN PRINT"{LEFT,SPACE}UN
     D","EIN SCHLUESSEL."                             <074>
8827 IF N(12)<>39 THEN PRINT                          <127>
8830 PRINT"VOM OBEREN RAND STARREN DICH EI
     NIGE HY- AENEN AN.":GOTO 4100                    <089>
8900 DATA -8,38                                       <111>
8910 IF FL%(6)=.THEN PRINT"DU BIST IN VOEL
     LIG UNDURCHDRINGLICHEM{3SPACE}GEBUESC
     H."                                              <192>
8920 IF FL%(6)THEN PRINT"DU STEHST IM GEST
     RAEUCH."                                         <050>
8930 GOTO 4100                                        <018>
9000 DATA -8,40                                       <018>
9010 PRINT"DU BIST UNTER EINEM ABGRUND.GER
     IPPE"                                            <165>
9020 PRINT"LIEGEN HERUM.":IF N(20)<>41 THE
     N 4100                                           <224>
9030 PRINT"EINE HOLZSTANGE LIEGT ZWISCHEN
     DEN KNO- CHEN.":GOTO 4100                        <025>
9100 DATA 0,0                                         <041>
9110 PRINT"DU HAENGST AM SEIL AN DER FELSW
     AND.":IF N(10)<>42 THEN 4100                     <231>
9120 PRINT"IM HORST SITZT EIN ADLER, EINIG
     E EIER{3SPACE}AUSBRUETEND.":GOTO 4100            <076>
9200 DATA ,                                           <165>
9210 PRINT"DU SITZT AUF DEM BAUM.NEBEN DIR
      IST DAS"                                        <100>
9220 PRINT"NEST DER ELSTER.":IF N(2)=43 TH
     EN PRINT"DARIN LIEGT EINE UHR."                  <069>
9230 IF FL%(15)=-1 THEN GOTO 4100                     <066>
9240 PRINT"EIN INDIANER TAUCHT UNTER DIR A
     UF, SIEHT";                                      <101>
9250 PRINT"UM SICH UND VERSCHWINDET WIEDER
     .":FL%(15)=-1:GOTO 4100                          <065>
9300 DATA ,                                           <011>
9310 PRINT"DU STEHST IN DEM DRECKIGEN WASS
     ER DES{3SPACE}BRUNNENS.":GOTO 4100               <177>
9400 DATA ,                                           <111>
9410 PRINT"DU BIST AUF EINER KLEINEN INSEL
     ."                                               <049>
9420 PRINT"DIE BLAETTER EINER STAUDE SPEND
     EN DIR{3SPACE}ANGENEHMEN SCHATTEN."              <124>
9425 IF N(5)=.THEN FL%(19)=9                          <045>
9430 IF N(21)<>45 THEN 4100                           <137>
9440 PRINT"DIE TRUHE STECKT HALB IM SAND."
     :GOTO 4100                                       <232>
9500 DATA 0,0                                         <189>
9510 PRINT"DU SITZT IN DER SPITZE DER STAU
     DE."                                             <211>
9520 IF N(5)=46 THEN PRINT"EIN BUSCH BANAN
     EN HAENGT ZUM GREIFEN{4SPACE}NAHE."              <130>
9530 GOTO 4100                                        <112>
9600 DATA -2,48                                       <018>
9610 PRINT"DU BIST IM SCHATZRAUM DES TEMPE
     LS!!!"                                           <062>
9620 IF FL%(13)THEN PRINT"DURCH EINEN SPAL
     T SCHEINT LICHT HEREIN."                         <177>
9625 IF FL%(13)THEN PRINT"VOR DEM SPALT IS
     T EINE NISCHE."                                  <157>
9630 IF N(16)>-1 THEN 4100                            <231>
9640 PRINT"DAS LICHT WIRD DURCH DIE FIGUR
     GEBROCHEN";                                      <109>
9650 PRINT"UND STRAHLT AUF EINE BESTIMMTE
     STELLE.":GOTO 4100                               <230>
9700 DATA 1,54,-2,52                                  <093>
9710 IF FL%(11)THEN PRINT"DIE SCHLANGEN ES
     SEN NOCH.":GOTO 4100                             <036>
9720 PRINT"ZWEI KOENIGSKOBRAS VERWEHREN DI
     R DEN{4SPACE}WEG.":GOTO 4100                     <031>
9800 DATA 1,52,-3,53                                  <084>
9810 IF FL%(12)THEN 9820                              <200>
9815 PRINT"DIE TUER SCHLIESST SICH HINTER
     DIR":FL%(12)=1                                   <197>
9820 PRINT"DU HAST EINEN FINSTEREN TEMPELR
     AUM BE-"                                         <171>
9830 PRINT"TRETEN.DER BODEN DES NACH NORDE
     N FUEHR-"                                        <071>
9835 PRINT"ENDEN GANGES IST HALB EINGESTUE
     RZT UND"                                         <082>
9837 PRINT"WIRD NUR NOCH DURCH EIN PAAR LO
     SE PLAN- KEN GEHALTEN."                          <195>
9840 PRINT"OESTLICH IST EINE FEINES SEIL G
     ESPANNT.":GOTO 4100                              <249>
9900 DATA -4,53                                       <029>
9910 PRINT"DU BIST IN EINEM DUNKLEN RAUM."
     :IF N(16)<>50 THEN 4100                          <198>
9920 PRINT"EINE GLAESERNE FIGUR SCHIMMERT
     GEHEIM-{2SPACE}NISVOLL.":GOTO 4100               <224>
10000 DATA -7,38                                      <181>
10010 GOTO 8610                                       <228>
10100 PRINT"DER GANG STUERZT UNTER DIR ZUS
      AMMEN.":GOTO 1500                               <087>
10200 PRINT"DAS SEIL ZERREISST, EIN STEIN
      LOEST SICH";                                    <237>
10210 PRINT"AUS DEM GEWOELBE UND BEGRAEBT
      DICH UNTERSICH.":GOTO 1500                      <239>
10300 IF N(10)=48 THEN P=47:GOTO 4000                 <085>
10310 PRINT"DAS HAETTEST DU NICHT DU NICHT
       SOLLEN!"                                       <054>
10320 PRINT"BLITZSCHNELL STOESST EINE DER
      SCHLANGEN"                                      <122>
10330 PRINT"VOR UND SCHLAEGT IHRE GIFTZAEH
      NE IN DEINFLEISCH.":GOTO 1500                   <087>
20000 :                                               <194>
20100 :                                               <008>
20200 :                                               <108>
20300 :                                               <210>
20400 :                                               <054>
20500 :                                               <156>
20600 :                                               <000>
20700 IF E(V)=0 THEN PRINT"DU KANNST NICHT
       IN DIESE RICHTUNG!":GOTO 4110                  <044>
20701 IF N THEN PRINT KN$:GOTO 4110                   <103>
20702 IF N(19)=-1 AND FL%(22)=.THEN PRINT
      KN$:GOTO 4110                                   <131>
20703 IF N(19)=-1 AND FL%(21)>-1 THEN PRIN
      T"MIT FALLSCHIRM ?":E=E(V):FL%(21)=2
      :GOTO 4110                                      <150>
20705 IF P<10 OR P>23 OR FL%(16)<>E(V)OR F
      L%(15)=-1 OR P=18 THEN 20710                    <010>
20707 PRINT"MIT EINEM MAL STEHT DIR EIN IN
```

**Listing »Inka« (Fortsetzung)**

```
        DIANER GE-";                              <140>
20708 PRINT"GENUEBER. ER GREIFT DICH AN UN
        D UEBER-{2SPACE}WAELTIGT DICH.":GOTO
        1500                                      <179>
20710 IF FL%(15)<>-1 THEN FL%(15)=3               <058>
20715 FL%(16)=P:IF P<>18 THEN P=E(V):GOTO
        4000                                      <044>
20720 PRINT"VON ALLEN SEITEN KOMMEN INDIAN
        ER HERBEI.";                              <113>
20725 PRINT"SIE SCHLEPPEN DICH WEG..."            <093>
20730 FOR I=1 TO 2000:NEXT:FL%(4)=5:P=34:G
        OTO 4000                                  <004>
20800 IF N=0 THEN PRINT WD$:GOTO 4110             <218>
20802 IF N(N)<>P THEN PRINT SN$:GOTO 4110         <018>
20805 IF O THEN 20850                              <012>
20810 GOSUB 20860:IF D THEN 1500                  <222>
20815 GOTO 4110                                   <253>
20850 PRINT"{3SPACE}"N$(N)" ODER "N$(O)" ?
        ":GOTO 4110                               <235>
20860 D=0:IF N>23 THEN PRINT KN$:RETURN           <051>
20861 IF N=7 AND P=8 AND FL%(1)=.THEN PRIN
        T"DER SKORPION STICHT DICH!":D=1:RET
        URN                                       <168>
20862 IF N=4 AND FL%(1)=1 THEN FL%(1)=.           <243>
20863 IF N=3 AND P=9 THEN PRINT"UNTER DEM
        STEIN LAG EIN DECKEL.":N(14)=9            <071>
20864 IF N=7 AND P=8 THEN N(36)=.                 <047>
20866 IF N=3 AND P=37 THEN FL%(5)=.               <165>
20870 Z=.:FOR I=1 TO 23:IF N(I)=.THEN Z=Z+
        1                                         <056>
20875 NEXT:IF Z>5 THEN PRINT"DU TRAEGST ZU
        VIEL!!":RETURN                            <108>
20880 N(N)=.:PRINT OK$:RETURN                     <168>
20900 IF N=.THEN PRINT WD$:GOTO 4110              <253>
20905 IF O<>.THEN GOTO 20920                      <097>
20907 IF N>23 THEN PRINT KN$:GOTO 4110            <128>
20910 GOTO 21000                                  <068>
20915 IF N=22 AND P=5 AND O=29 THEN 21012         <234>
20920 IF NOT(N=11 AND P=34 AND O=33 AND N(
        11)=.AND FL%(4)<>.)THEN 20930             <033>
20925 PRINT"SIE LASSEN DICH IN FRIEDEN UND
        GEHEN.":N(11)=-1:FL%(4)=0:GOTO 4110       <034>
20930 IF NOT(N=46 AND O=5 AND N(46)=P)THEN
        20950                                     <058>
20935 IF N(5)THEN PRINT HN$:GOTO 4110             <219>
20937 PRINT"DER AFFE STUERZT SICH AUF DIE
        BANANE.":FL%(18)=4:FL%(19)=FL%(19)-1      <173>
20940 GOTO 4110                                   <122>
20950 IF NOT((N=33 OR N=46 OR N=37)AND O<2
        4)THEN PRINT KN$:GOTO 4110                <013>
20960 PRINT"DER ";N$(N);" SCHAUT DARAUF,":
        PRINT"UND WIRFT ES HIN.":N(O)=P:GOTO
        4110                                      <157>
21000 IF N=13 THEN 21050                          <060>
21007 IF N=19 AND N(N)<.THEN N(N)=.               <216>
21010 IF N(N)<>.THEN PRINT HN$:GOTO 4110          <239>
21011 IF NOT(N=22 AND P=5 AND N(4)=5)THEN
        21013                                     <145>
21012 PRINT"ER SAGT:<HALT DICH AM EIMER FE
        ST!>":FL%(7)=1:N(22)=-1:GOTO 4100         <198>
21013 IF NOT(N=2 AND P=9)THEN 21015               <046>
21014 PRINT"DER EINSIEDLER STUERZT SICH AU
        F DIE UHR.":FL%(2)=3:N(2)=-1:N(3)=9:
        GOTO 4110                                 <237>
21015 IF NOT(N=3 AND(P=37 OR P=51))THEN 21
        017                                       <065>
21016 PRINT"DU KANNST JETZT UEBER DEN FLUS
        S.":FL%(5)=1:N(3)=37:GOTO 4110            <199>
21017 IF P=34 AND N=11 THEN O=33:GOTO 2092
        0                                         <182>
21018 IF P=N(46)AND N=5 THEN 20935                <040>
21020 IF NOT(N=16 AND O=47 AND P=47 AND FL
        %(13)=1)THEN 21029                        <078>
21022 PRINT"DIE FIGUR PASST SICH IN DIE NI
        SCHE EIN.":N(16)=-1                       <085>
21023 PRINT"EIN LICHTSTRAHL WEIST AUF EINE
        STELLE{3SPACE}DES BODENS.":GOTO 411
        0                                         <216>
21029 IF O THEN 20850                             <238>
21030 N(N)=P:PRINT OK$:GOTO 4110                  <125>
21050 FOR I=1 TO 23:IF N(I)=.THEN N(I)=P:P
        RINT OK$                                  <032>
21060 NEXT:GOTO 4110                              <245>
21100 IF N THEN PRINT KN$:GOTO 4110               <250>
21105 IF P=28 THEN P=45:GOTO 4000                 <206>
21106 IF P<>38 THEN 21110                         <144>
21107 FOR I=1 TO 23:IF N(I)=.THEN N(I)=-2         <242>
21108 NEXT:P=3:GOTO 4000                          <108>
21110 IF P<>45 THEN 21120                         <099>
21115 IF FL%(5)THEN PRINT"DU ZERSCHELLST A
        N EINEM STEIN IM FLUSS.":GOTO 1500        <227>
21118 P=3:GOTO 4000                               <062>
21120 IF P<>3 THEN PRINT KN$:GOTO 4110            <119>
21125 PRINT"DU WIRST DEN WASSERFALL HINUNT
        ER GE-{4SPACE}SCHWEMMT.":GOTO 1500        <106>
21200 IF N=4 AND(O=.OR O=35)THEN 21220            <163>
21210 PRINT KN$:GOTO 4110                         <211>
21220 IF N(4)THEN PRINT HN$:GOTO 4110             <218>
21225 PRINT"DER SKORPION IST GEFANGEN."           <139>
21230 N(4)=8:FL%(1)=1:GOTO 4110                   <088>
21300 IF O THEN 20850                             <255>
21305 IF N=.THEN PRINT WD$:GOTO 4110              <150>
21310 IF N<>15 OR P<>39 OR N(15)<>P THEN P
        RINT KN$:GOTO 4110                        <197>
21315 N(15)=-1                                    <204>
21316 IF FL%(8)+FL%(9)=2 THEN 21320               <108>
21317 PRINT"DER KADAVER WIRD VON DEN HYAEN
        EN SOFORT"                                <155>
21318 PRINT"WEGGESCHLEPPT UND GEFRESSEN.":
        GOTO 4110                                 <246>
21320 PRINT"DIE HYAENEN STUERZEN SICH GIER
        IG AUF DEN";                              <208>
21330 PRINT"KADAVER UND ZERREN DICH DARAN
        AUS DER{3SPACE}GRUBE.":P=38:GOTO 400
        0                                         <159>
21400 IF(N=.OR N=31 AND O=.)AND P=25 THEN
        P=28:GOTO 4000                            <233>
21410 IF N THEN PRINT KN$:GOTO 4110               <050>
21412 IF P=43 THEN P=16:GOTO 4000                 <253>
21415 IF P=46 THEN P=45:GOTO 4000                 <197>
21417 IF NOT(P=5 OR P=34 OR P=31 OR P=32)T
        HEN PRINT KN$:GOTO 4110                   <104>
21418 IF N(19)>-1 THEN PRINT"DU ZERSCHELLS
        T...":GOTO 1500                           <249>
21419 IF P=31 THEN PRINT"DEIN FALLSCHIRM B
        LEIBT HAENGEN UND ZER- REISST...":GO
        TO 1500                                   <133>
21420 IF P=34 THEN PRINT"ES GEHT ABWAERTS.
        ..":P=41:FL%(22)=1:GOTO 4000              <088>
21421 IF P=32 THEN PRINT"DU SCHWEBST SANFT
        HERAB!":P=35:FL%(22)=1:GOTO 4000          <212>
21425 PRINT"DU FAELLST IN DEN BRUNNEN...":
        GOTO 1500                                 <028>
21500 IF N<>8 OR P<10 OR P>23 OR P=18 THEN
        21530                                     <029>
21510 IF N(1)THEN PRINT KN$:GOTO 4110             <166>
21520 N(8)=P:N=8:GOTO 20810                       <225>
21530 IF N<>45 OR P<>40 THEN PRINT KN$:GOT
        O 4110                                    <157>
21540 IF N(1)THEN PRINT KN$:GOTO 4110             <196>
21550 PRINT"DU HAST DAS GEBUESCH WEGBAR GE
        MACHT.":FL%(6)=1:GOTO 4110                <070>
21700 GOTO 4000                                   <072>
22100 IF O THEN 20850                             <037>
22105 IF N=.THEN PRINT WD$:GOTO 4110              <188>
22110 IF(N=4 AND(P=44 OR P=5)AND N(4)=5)OR
        (N=8 AND P=39 AND N(15)>.)THEN 22130      <138>
22120 PRINT KN$:GOTO 4110                         <105>
22130 IF N=8 THEN FL%(8)=1:PRINT OK$:GOTO
        4110                                      <211>
22140 IF FL%(7)=.THEN PRINT"DU MUSST IHN B
        EZAHLEN!":GOTO 4110                       <127>
22150 PRINT"ER KURBELT KRAEFTIG AN DER WIN
        DE.":P=49-P:GOTO 4000                     <172>
22200 Z=.:PRINT"DU HAST:":FOR I=1 TO 23:IF
        N(I)=.THEN PRINT"-"N$(I):Z=1              <124>
22210 NEXT:IF Z=.THEN PRINT"NICHTS"               <178>
22220 GOTO 4110                                   <132>
22400 IF N=.THEN PRINT WD$:GOTO 4110              <229>
22410 IF NOT(N=28 AND P=2 AND FL%(.)=1)THE
        N PRINT KN$:GOTO 4110                     <110>
22420 PRINT"DU HOERST EIN SELTSAMES,SCHNAR
        RENDES GE-RAEUSCH.":FL%(13)=1             <138>
22430 GOTO 4110                                   <088>
22600 IF O THEN 20850                             <029>
22610 IF N=43 AND P=7 THEN P=8:GOTO 4000          <127>
22615 IF N=30 AND P=1 THEN P=2:GOTO 4000          <010>
22620 IF N=31 AND P=3 THEN PRINT"DU WIRST
        DEN WASSERFALL HINABGESPUELT...":GOT
        O 1500                                    <153>
22625 IF NOT(N=32 AND(P=4 OR P=38 OR P=37
        OR P=51))THEN 22630                       <045>
22626 IF P=4 OR P=38 THEN P=-37*(P=4)-51*(
```

```
      P=38):GOTO 4000                          <162>
22627 IF FL%(5)=.THEN PRINT"DU WIRST WEGGE
      SPUELT...":GOTO 1500                     <032>
22628 P=88-P:GOTO 4000                         <045>
22630 IF N=40 AND P=36 AND FL%(14)=1 THEN
      P=49:GOTO 4000                           <041>
22635 IF N=41 AND P=47 AND FL%(13)=1 THEN
      P=2:GOTO 4000                            <236>
22640 IF N=48 AND(P=26 OR P=24)THEN P=25:G
      OTO 4000                                 <002>
22645 IF N=49 AND P=29 THEN P=30:GOTO 4000     <042>
22650 IF N=45 AND P=40 AND FL%(6)=1 THEN P
      =41:GOTO 4000                            <132>
22660 PRINT KN$:GOTO 4110                      <137>
22700 IF N=16 AND O=47 AND P=47 AND FL%(13
      )=1 THEN 21022                           <052>
22710 PRINT KN$:GOTO 4110                      <187>
22800 IF O THEN 20850                          <231>
22820 IF(P<>49 AND P<>50)OR N<>51 THEN PRI
      NT KN$:GOTO 4110                         <108>
22830 P=99-P:GOTO 4000                         <041>
23000 IF N THEN PRINT KN$:GOTO 4110            <116>
23010 IF N(18)THEN PRINT"ICH HABE KEINE SC
      HAUFEL...":GOTO 4110                     <164>
23015 IF P<>45 THEN 23030                      <242>
23020 PRINT"DU SIEHST EINE TRUHE, NOCH HAL
      B IM SAND.";:N(21)=45:GOTO 4110          <027>
23030 IF P<>47 THEN PRINT KN$:GOTO 4110        <066>
23040 IF N(16)<>-1 THEN PRINT"WO SOLL ICH
      DENN GRABEN ???":GOTO 4110               <225>
23050 PRINT"DU GRAEBST DEN INKASCHATZ AUS!
      ":N(16)=-2:N(17)=47:GOTO 4110            <138>
23200 IF N=.THEN PRINT WD$:GOTO 4110           <011>
23202 IF O THEN 20850                          <123>
23204 IF NOT(P=11 AND N=7 AND FL%(10)=.AND
      N(36)=.)THEN 23210                       <191>
23205 IF N(14)THEN PRINT"OHNE DECKEL ?":GO
      TO 4110                                  <213>
23206 IF N(7)THEN PRINT HN$:GOTO 4110          <012>
23207 FL%(10)=1:N(14)=-1:N(36)=-1:PRINT"DI
      E BIENEN SIND GEFANGEN.":GOTO 4110       <039>
23210 IF N=7 THEN PRINT"DAS KANN ICH JETZT
      NICHT.":GOTO 4110                        <163>
23215 IF N=40 AND P=36 THEN PRINT OK$:FL%(
      14)=.:GOTO 4110                          <007>
23220 PRINT KN$:GOTO 4110                      <189>
23300 IF N=.THEN PRINT WD$:GOTO 4110           <113>
23310 IF NOT(N=7 AND FL%(10)=1)OR O THEN 2
      3330                                     <073>
23312 IF N(7)THEN PRINT HN$:GOTO 4110          <120>
23315 N(14)=.:IF P<>2 THEN PRINT"DIE BIENE
      N FLIEGEN DAVON.":FL%(10)=.:GOTO 411
      0                                        <168>
23320 PRINT"DIE BIENEN VERJAGEN DEN INDIAN
      ER.":FL%(.)=1:GOTO 4110                  <066>
23330 IF NOT(N=40 AND P=36)THEN 23340          <223>
23331 IF N(20)THEN PRINT HN$:GOTO 4110         <166>
23332 IF O=20 THEN 23336                       <050>
23333 PRINT"ALS DU DAS PORTAL AUFDRUECKST,
      STICHT"                                  <221>
23334 PRINT"DICH ETWAS IN DIE HAND.DIR WIR
      D SCHWIN-"                               <205>
23335 PRINT"DELIG UND AUGENBLICKE SPAETER
      KIPPST DU TOT UM.":GOTO 1500             <226>
23336 IF N(12)THEN PRINT"DIR FEHLT DER PAS
      SENDE SCHLUESSEL.":GOTO 4110             <109>
23337 PRINT"DAS GROSSE PORTAL SCHWINGT SAN
      FT AUF...":FL%(14)=1:GOTO 4110           <153>
23340 IF N<>21 THEN PRINT KN$:GOTO 4110        <232>
23350 IF N(12)THEN PRINT"ICH HABE KEINEN S
      CHLUESSEL...":GOTO 4110                  <154>
23360 PRINT"DIE TRUHE IST LEER.":GOTO 4110     <222>
23400 IF O=.THEN PRINT WD$:GOTO 4110           <221>
23410 IF NOT(N=8 AND P=39 AND O=15 AND N(1
      5)>.)THEN 23450                          <182>
23430 IF N(8)THEN PRINT HN$:GOTO 4110          <015>
23440 PRINT OK$:FL%(9)=1:N(8)=-1:GOTO 4110     <112>
23450 IF NOT(N=8 AND O=44 AND P=31)THEN 23
      470                                      <010>
23455 IF N(8)THEN PRINT HN$:GOTO 4110          <040>
23460 PRINT OK$:FL%(17)=1:N(8)=31:GOTO 411
      0                                        <000>
23470 PRINT KN$:GOTO 4110                      <185>
23600 IF O THEN 20850                          <013>
23605 IF N=.AND(P=49 OR P=48)THEN P=97-P:G
      OTO 4000                                 <227>
23608 IF N=.THEN PRINT WD$:GOTO 4110           <167>
```

```
23610 IF N=24 AND P=16 THEN P=43:GOTO 4000     <222>
23615 IF N=50 AND P=45 THEN P=46:GOTO 4000     <073>
23618 IF N=8 AND(P=31 OR P=42)AND FL%(17)=
      1 THEN P=73-P:GOTO 4000                  <227>
23620 PRINT KN$:GOTO 4110                      <081>
23700 IF N THEN PRINT KN$:GOTO 4110            <054>
23710 IF P=49 OR P=48 THEN 23605               <228>
23720 PRINT KN$:GOTO 4110                      <181>
23800 IF N<>11 OR P<>27 THEN PRINT KN$:GOT
      O 4110                                   <136>
23802 IF O THEN 20850                          <215>
23805 IF N(23)THEN PRINT"MIR FEHLT DIE AUS
      RUESTUNG.":GOTO 4110                     <230>
23810 PRINT"DU KANNST TATSAECHLICH GOLD WA
      SCHEN!":N(11)=27:GOTO 4110               <138>
23900 IF(N=39 AND(O=.OR O=36))OR(N=36 AND(
      O=.OR O=39))THEN 23930                   <099>
23920 PRINT KN$:GOTO 4110                      <127>
23930 IF N(36)THEN PRINT HN$:GOTO 4110         <162>
23935 PRINT"DER AST WIRD KLEBRIG.NEUGIERIG
       SETZT SI-";                             <048>
23940 PRINT"CH DIE ELSTER AUF IHN UND KLEB
      T FEST!":FL%(3)=1:GOTO 4110              <026>
24300 INPUT "{DOWN}FILENAME ";FI$              <154>
24310 PRINT"{RVSON}D{RVOFF}ISK ODER{SPACE,
      RVSON}T{RVOFF}APE ?"                     <180>
24315 GET IN$:IF IN$=""THEN 24315              <187>
24320 IF IN$="D"THEN OPEN 1,8,2,FI$+",S,W"
      :GOTO 24330                              <071>
24325 IF IN$="T"THEN OPEN 1,1,1,FI$:GOTO 2
      4330                                     <099>
24328 GOTO 24315                               <012>
24330 PRINT#1,P:FOR I=1 TO MX:PRINT#1,N(I)
      :NEXT                                    <168>
24340 FOR I=0 TO FL:PRINT#1,FL%(I):NEXT        <208>
24350 CLOSE 1:PRINT OK$:GOTO 4110              <217>
24400 INPUT "{DOWN}FILENAME ";FI$              <000>
24410 PRINT"{RVSON}D{RVOFF}ISK ODER{SPACE,
      RVSON}T{RVOFF}APE ?"                     <026>
24415 GET IN$:IF IN$=""THEN 24415              <035>
24420 IF IN$="D"THEN OPEN 1,8,2,FI$+",S,R"
      :GOTO 24430                              <152>
24425 IF IN$="T"THEN OPEN 1,1,0,FI$:GOTO 2
      4430                                     <200>
24428 GOTO 24415                               <144>
24430 INPUT#1,P:FOR I=1 TO MX:INPUT#1,N(I)
      :NEXT                                    <178>
24440 FOR I=0 TO FL:INPUT#1,FL%(I):NEXT        <228>
24450 CLOSE 1:PRINT OK$:GOTO 4000              <013>
24500 PRINT"{2DOWN}WIRKLICH ?"                 <053>
24510 GET IN$:IF IN$=""THEN 24510              <092>
24520 IF IN$="J"THEN PRINT"{CLR}GOTO4000:[
      ZUM WEITERMACHEN]":POKE 631,141:END      <112>
24530 IF IN$="N"THEN POKE 198,0:GOTO 4110      <039>
24540 GOTO 24510                               <158>
24800 IF P<>6 OR N<>34 THEN PRINT KN$:GOTO
       4110                                    <147>
24810 IF O=.THEN PRINT"WOMIT DENN ?":GOTO
      4110                                     <065>
24820 IF O<>6 THEN PRINT KN$:GOTO 4110         <192>
24825 IF N(6)THEN PRINT HN$:GOTO 4110          <075>
24830 PRINT"EIN FEUER KOMMT AUF.JETZT HEIS
      ST ES WAR-";                             <121>
24840 PRINT"TEN.":FOR I=1 TO 2000:NEXT         <154>
24850 PRINT"EIN HUBSCHRAUBER KOMMT UND LAE
      SST EINE"                                <002>
24855 PRINT"STRICKLEITER HERUNTER, AN DER
      DU HOCHKL-";                             <051>
24860 PRINT"ETTERST."                          <198>
24870 FOR I=1 TO 2000:NEXT:GOTO 40000          <168>
25000 IF N THEN PRINT KN$:GOTO 4110            <084>
25010 PRINT OK$:GOTO 4110                      <185>
25100 IF N<>19 OR N(19)THEN PRINT KN$:GOTO
       4110                                    <037>
25110 N(19)=-1:PRINT OK$:GOTO 4110             <182>
25300 IF N THEN PRINT KN$:GOTO 4110            <130>
25310 IF FL%(21)THEN PRINT"OKAY, WAR NUR '
      NE FRAGE...":P=E:FL%(21)=-1:GOTO 400
      0                                        <127>
25320 PRINT"WAS SOLL DAS DENN ?":GOTO 4110     <177>
30000 DATA 0,0,0,0,0,0,0,0,2,0,0,15            <086>
30010 DATA 2,217,255,15,255,190,31,255         <006>
30020 DATA 252,63,255,248,127,255,252          <042>
30030 DATA 127,255,254,125,255,255,63          <133>
30040 DATA 127,199,15,188,224,3,206            <166>
30050 DATA 112,0,0,0,0,0,0,0,0,0,0,0           <108>
30060 DATA 0,0,0,0,0,0,0,0,0,0                 <253>
```

```
30100 DATA 0,255,0,7,255,224,31,255          <211>
30101 DATA 248,127,255,254,127,255,254       <197>
30102 DATA 255,255,255,255,255,255,255       <015>
30103 DATA 129,255,240,129,15,200,66         <188>
30104 DATA 19,132,36,33,194,0,67,49          <035>
30105 DATA 60,140,12,60,48,3,24,192          <058>
30106 DATA 0,126,0,0,189,0,0,60,0,0          <160>
30107 DATA 60,0,0,36,0,0,102,0               <133>
30110 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0         <015>
30111 DATA 0,0,255,255,0,1,128,0,15          <118>
30112 DATA 240,0,51,248,0,71,248,3,143       <175>
30113 DATA 252,7,255,255,255,255,255         <110>
30114 DATA 255,127,248,0,16,32,0,127         <135>
30115 DATA 248,0,0,0,0,0,0,0,0,0,0,0         <166>
30116 DATA 0,0,0,0,0                         <197>
40000 FOR I=1 TO 1000:NEXT                   <036>
40010 PRINT"{CLR}":PRINT"DU BIST WIEDER IM
       SICHEREN HUBSCHRAUBER!";              <134>
40020 IF N(17)=.THEN PRINT"{DOWN}DU HAST D
      EN SCHATZ GEFUNDEN !"                  <037>
40030 IF N(9)=.THEN PRINT"{DOWN}DU HAST EI
      N ZEICHEN DES FORSCHERS GE-{3SPACE}B
      RACHT !"                               <113>
40035 IF N(17)THEN PRINT"{DOWN}ABER WO IST
       DER SCHATZ ?"                         <252>
40038 IF N(9)THEN PRINT"{DOWN}UND WO IST E
      IN ZEICHEN DES FORSCHERS ?"            <178>
40040 IF N(17)OR N(9)THEN PRINT"{3DOWN,RVS
      ON}MISSION NICHT ERFUELLT !!!":GOTO
      40100                                  <087>
40050 PRINT"{2DOWN}MISSION ERFUELLT !!!{DO
      WN}"                                   <079>
40060 R=18:FOR I=1 TO 500:PRINT CHR$(R)"HI
      R{SHIFT-SPACE}GRATULIEREN!!{UP}":R=1
      64-R:NEXT                              <098>
40100 PRINT"{2DOWN}AUFHOEREN ?":GOTO 24510 <119>
```

**Listing »Inka« (Schluß)**

# Wie im Eisen der Fuchs...

**Mord! Der Gastgeber eines politischen Kongresses fiel einem Anschlag zu Opfer. Wer war der Täter? Klären Sie das Verbrechen auf.**

Ein Wort am Anfang. In diesem Sonderheft ist zu diesem Adventure kein Listing abgedruckt. Das liegt ganz einfach daran, daß das Programm aus über 90 Einzelteilen besteht, die etwa 25 Seiten Listing ergeben würden. Diese Tipparbeit wollten wir Ihnen nicht zumuten. Deshalb ist das Programm nur auf der Leser-Service-Diskette zu diesem Heft erhältlich.

Das Adventure veralbert auf leise Weise den charakteristischen Stil englischer Kriminalromane, den Agatha Christie bis zur Perfektion und viele Roman-Leser bis an den Rand der Verzweiflung getrieben hat. In den 20er Jahren des 20. Jahrhunderts findet in Falmouth in der englischen Grafschaft Cornwall eine politische Tagung statt. Die dabei teilnehmende Gruppe strebt eine Verlängerung des Pachtvertrages der Kronkolonie Hongkong an, der gegen Ende dieses Jahrtausends auslaufen soll. Es nehmen folgende Personen an der Konferenz teil:

- Lawrence Templeton-Douglas, der Hausherr;
- Kriegsveteran Dr. Simon Frank, Vertreter des Unterhauses;
- Oliver Bell, Duke of Cheeseborough, Vertreter des Oberhauses;
- Cathrine de Vidocq, Vertreterin Frankreichs;
- Joseph »Santiago« Shulz, amerikanischer »Unternehmer«;
- Viktor von Stein, Vertreter Deutschlands;
- Sid Wallace, Agent des amerikanischen FBI;
- Gerald C. Green, Gärtner;
- Gung Ho, Koch des Hauses;
- George Aston-Peacock, Butler.

Im Laufe der Tagung fällt der Hausherr einem Anschlag einer nicht näher zu benennenden chinesischen Untergrundgruppe zum Opfer. Um den Fall nicht an die Öffentlichkeit zu tragen, wurden Sie, Jonathan Edwards, Angehöriger des Secret Service, beauftragt, den Täter, der sich unter den Konferenzteilnehmern befinden muß, zu enttarnen. Nicht an der Konferenz teilnehmende Personen können ausgeschlossen werden, da das Gelände hermetisch abgeriegelt ist. Ihnen steht nur ein Handbuch zur Verfügung, das die Fingerabdrücke aller oben genannten Personen enthält.

Die Bedienung des Programms wird durch die Belegung der Funktionstasten mit oft gebrauchten Befehlen erleichtert.

F1 und F3 dienen zu Diskettenoperationen. So speichert F1 einen Spielstand auf Diskette. Ein Spielstand gleichen Namens, der sich eventuell auf ihr befindet, wird vorher gelöscht. Die Länge des Filenamens darf maximal 10 Zeichen umfassen.

F3 ist das genaue Gegenstück dazu. Hiermit wird ein Spielstand geladen. Dabei ist es zwingend erforderlich, daß sich dieser auf derselben Diskettenseite befindet, auf der die Bilder und Texte der Räume enthalten sind.

Taste F5 gibt eine Liste der gültigen Befehle und deren Synonyme aus.

F7 entspricht dem in vielen Adventures verwendeten Befehl »Inventory«. Alle Gegenstände, die der Spieler bei sich trägt, werden angezeigt.

Hier noch ein Tip zu den Gegenständen: es können nicht unbegrenzt viele getragen werden, und es ist nicht alles Gold, was glänzt.

RUN/STOP startet das Spiel nach einer erfolgten Sicherheitsabfrage neu. Dies ist beispielsweise nötig, wenn Sie in eine der wenigen, aber tückischen Fallen des Täters laufen. Wird RETURN ohne Eingabe eines Kommandos gedrückt, so werden noch einmal Bild und Text des aktuellen Raumes eingeladen.

Das Spiel umfaßt 81 Räume mit eigenem Text und Grafik, neun eigenständigen Charakteren und einen Sprachschatz von 156 verarbeitbaren Worten. (Olaf Barthel/dm)

# Spion III – die Jagd nach der Bombe

**Versuchen Sie, das teuerste Projekt in der Geschichte der bemannten Raumfahrt vor Sabotageakten eines gewinnsüchtigen Geschäftskonzerns zu schützen.**

Soeben hebt in Florida das Space Shuttle mit dem europäischen Raumlabor Spacelab ab. Die Mission »D-2« beginnt. Alle Vorzeichen sprechen für einen herausragenden Erfolg dieses Unternehmens. An Bord findet ein Experiment statt, das im Falle eines Erfolges die Herstellung von Insektenvertilgungsmitteln überflüssig machen würde. Dies könnte jedoch den Ruin des marktbeherrschenden Konzerns bedeuten. Der Bundesnachrichtendienst (BND) verfügt aber über Informationen, nach welchen das Unternehmen einen Sprengsatz an Bord geschmuggelt hat. Sie als Spezialagent bekommen nun die Aufgabe übertragen, in das Kontrollzentrum dieses Konzerns einzudringen und den Countdown bis zur Sprengung zu stoppen.

Das Ladeprogramm (Listing 1):

Hier wird der Basic-Start um 2 KByte nach oben verschoben, um später für die Grafikdaten Platz zu schaffen. Da hierdurch das Ladeprogramm gelöscht wird, muß man dieses vor dem ersten Start unbedingt speichern. Außerdem wird hier ein Musikprogramm eingelesen, das später interrupt-gesteuert ein Lied von J. S. Bach spielt. Das Unterprogramm, in reiner Maschinensprache geschrieben, arbeitet dreistimmig. Die Routine beginnt ab Adresse $C400. Die Daten für die drei Stimmen befinden sich in den Speicherbereichen ab $C600, $C900 und $CC00.

Das Hauptprogramm (Listing 2):

Zuerst einmal wird das Titelbild ausgegeben, die Musik aktiviert und eine Interruptroutine eingelesen. Während man nun das Titelbild betrachtet, werden die Grafikdaten eingelesen und die Strings initialisiert, durch die die Bilder dargestellt werden. Außerdem erhalten diese zu jedem Bild eine kurze Information. Ab Zeile 1000 folgen dann Unterprogramme, die später die Grafik ausdrucken. Nach 9000 werden noch einmal DATAs eingelesen. Diese dienen zur Erstellung eines Rasterzeilen-Interrupts und zur Realisierung des Eingabe-Scrollings. Jetzt werden noch die Eingabevokabeln definiert und die Hauptschleife ab 10 000 gestartet, die die Befehlseingabe in zwei Worte zerlegt (Verb und Objekt). Die hierbei verwendetet INPUT-Routine findet man ab Zeile 51 000.

Danach folgt die Auswertung der eingegebenen Befehle. Diese nimmt in der Hauptschleife den meisten Raum ein. Ab der Zeile 20 000 wird das Bild aufgebaut, indem die für die Bilder spezifischen Unterprogramme durch ON – GOTO aufgerufen werden.

Die neue INPUT-Routine besteht hauptsächlich aus einer Schleife, in der mittels GET-Anweisung die Zeichen von der Tastatur eingelesen werden. Der Vorteil der neuen Eingabe besteht in der Begrenzung der erlaubten CHR$-Codes. So kann man nun die Bildschirmmaske nicht mehr wegscrollen.

Die Eingaben bestehen normalerweise (außer bei Richtungsangaben etc.) aus zwei Worten: einem Verb (oder sonstigem Befehlswort) und einem Objekt. Sollten mehr als zwei Worte eingegeben werden, so erfolgt eine entsprechende Fehlermeldung.

Das Spiel versteht 26 Wörter. Diese lauten:

N,O,S,W,H (hoch), R (runter) ZEIT, PUNKTE (die Zeit ergibt sich aus der Anzahl der eingegebenen gültigen Befehle)

NIMM, VERLIERE (das Objekt »ALLES« funktioniert)

INV (Inventur)

STIRB (beendet das Spiel durch einen RUN/STOP – RESTORE

BETRITT, VERLASSE (das Objekt ist unbedingt anzugeben)

SAVE, LOAD, LIES, WERTE, GRABE, SCHALTE, ZÜNDE, ISS, SCHLIESSE, BETE, DRÜCKE, ÖFFNE sowie noch einige unwichtige, die nur einen Kommentar erzeugen.

Zu beachten wäre noch, daß die Eingabe der Befehle im Groß-/Kleinschriftmodus erfolgen muß. Außerdem ist es zwingend erforderlich, das Hauptprogramm (Listing 2) unter dem Namen »D.2« zu speichern.

(Steffen Goebbels/dm)

**Listing 1. Das Ladeprogramm zu »Spion III«**

```
200 POKE 53280,0:POKE 53281,0:PRINT"{CLR,W
    HITE}":POKE 53272,23                     <180>
210 PRINT TAB(8)"SPION III (D 2 MISSION)"   <195>
220 PRINT TAB(9)"{DOWN,CYAN}VON STEFFEN GO
    EBBELS"                                  <043>
230 PRINT TAB(15)"{DOWN,WHITE}(C) 1985"     <237>
240 PRINT TAB(7)"{BLUE,3DOWN}CA. 1:15 MIN.
     LADEZEIT."                              <168>
250 PRINT"{DOWN,YELLOW}BITTE LASSEN SIE DI
    E FLOPPY ANGESCHALTET{DOWN}"             <161>
260 PRINT"{BLACK}POKE4096,0:POKE44,16:NEW" <198>
270 PRINT"{2DOWN}LOAD"CHR$(34)"D.2"CHR$(34
    )",8{7UP}"                               <052>
280 POKE 631,13:POKE 632,13:POKE 633,82:PO
    KE 634,85:POKE 635,78:POKE 636,13:POKE
     198,6                                   <222>
900 REM *** SID INITIALISIEREN ***          <230>
910 S=54272:FOR I=S TO S+24:POKE I,0:NEXT   <134>
930 POKE 53245,1:POKE 53246,1:POKE S+5,96:
    POKE S+12,9:POKE S+19,0                  <203>
940 POKE 53247,1:POKE S+2,0:POKE S+9,255:P
    OKE S+16,0:POKE S+13,140:POKE S+20,240 <085>
950 POKE S+6,100                             <087>
999 REM ** MASCHINENPRG FUER MUSIK **       <122>
1000 DATA 174,255,207,202,142,255,207,240,
     21,174,254,207,202,142,254,207,240,83 <039>
1010 DATA 174,253,207,202,142,253,207,240,
     71,76,49,234,32,66,196,169,0,141,4,21
     2                                      <219>
1020 DATA 173,250,207,174,251,207,172,252,
     207,141,1,212,142,0,212,140,255,207   <170>
1030 DATA 234,234,169,33:REM←WELLENF.S1    <135>
1040 DATA 141,4,212,76,9,196,162,0,189,0,1
     98,157,250,207,232,224,3,208,245,24   <237>
1050 DATA 173,69,196,105,3,141,69,196,144,
     7,174,70,196,232,142,70,196,96,76,167 <023>
1060 DATA 196,32,135,196,169,0,141,11,212,
     173,250,207,174,251,207,172,252,207   <224>
1070 DATA 141,8,212,142,7,212,140,254,207,
     169,65:REM←WELLENF.S2                 <157>
1080 DATA 141,11,212,76,18,196,162,0,189,0
     ,201,157,250,207,232,224,3,208,245,24 <067>
1090 DATA 173,138,196,105,3,141,138,196,14
     4,7,174,139,196,232,142,139,196,96    <129>
1100 DATA 32,204,196,169,0,234,141,18,212,
     173,250,207,174,251,207,172,252,207   <042>
1110 DATA 141,15,212,142,14,212,140,253,20
     7,240,40,169,17:REM←WELLENF.S3        <240>
1120 DATA 141,18,212,76,49,234,162,0,189,0
```

**Listing 1. (Fortsetzung)**

```
     ,204,157,250,207,232,224,3,208,245,24 <012>
1130 DATA 173,207,196,105,3,141,207,196,14
     4,7,174,208,196,232,142,208,196,96    <131>
1140 DATA 169,0,141,69,196,141,138,196,141
     ,207,196,169,198,141,70,196,169,201   <191>
1150 DATA 141,139,196,169,204,141,208,196,
     169,64,141,255,207,141,254,207,141,25
     3                                     <088>
1160 FOR I=50176 TO 50464:READ Q:POKE I,Q:
     NEXT:REM SYS50441                     <121>
1170 DATA 207,76,49,234,120,169,0,141,20,3
     ,169,196,141,21,3,88,96               <079>
1180 REM **** STIMME 1 ****                <105>
2000 DATA 7493,64,6676,64,7493,64,8411,32,
     8911,32,10002,64,8911,128,8411,64     <008>
2010 DATA 8411,64,10002,64,10002,64,7493,6
     4,8411,64,6676,128,5947,64,5001,64    <063>
2020 DATA 6676,64,7493,64,8411,64,8411,64,
     7493,128,0,64,8411,64,6676,64,7493,64 <122>
2030 DATA 8411,32,8911,32,10002,64,8911,12
     8,8411,64,8411,64,10002,64,10002,64   <212>
2040 DATA 7493,64,8411,64,6676,128,6301,64
     ,5001,64,6676,64,7493,64,8411,64      <136>
2050 DATA 8911,64,8411,64,7493,64,6676,64,
     0,0                                   <171>
2999 I=50688                               <193>
3000 FOR C=0 TO 91 STEP 2:READ Q:READ Q1:P
     OKE I+2,Q1:HI=INT(Q/256)              <195>
3001 LO=Q-(HI*256):POKE I,HI:POKE I+1,LO:I
     =I+3:NEXT                             <080>
3010 REM **** STIMME 2 :  ****             <105>
3020 DATA 5001,64,5613,64,6301,64,6676,32,
     7493,32,8411,64,6676,64,7493,64,8411  <012>
3030 DATA 64,6676,64,6301,64,6676,64,6301,
     64,6676,32,6301,32,5613,128,5001,64   <167>
3040 DATA 6301,64,5001,64,5613,32,6301,32,
     6676,64,6676,64,6301,128,0,64,6301,64 <057>
3050 DATA 6676,64,6301,32,5613,32,5001,64,
     8411,64,8411,64,7493,64,7072,64,6676  <105>
3060 DATA 64,6301,64,6676,32,6301,32,5613,
     64,6301,64,6676,32,6301,32,5613,64    <232>
3070 DATA 5001,64,5001,64,5001,32,5613,32,
     6301,64,6676,64,7072,64,6676,64,6301  <212>
3080 DATA 64,6676,64,0,0                   <109>
3999 I=51456                               <205>
4000 FOR C=0 TO 107 STEP 2:READ Q:READ Q1:
     POKE I+2,Q1:HI=INT(Q/256)             <214>
4001 LO=Q-(HI*256):POKE I,HI:POKE I+1,LO   <235>
4002 I=I+3:NEXT                            <206>
4010 REM **** STIMME 3 :  ****             <121>
4020 DATA 3338,64,4455,64,4205,32,3764,32,
     3338,64,6676,64,5613,64,6301,64,6676  <129>
4030 DATA 64,5613,64,4205,32,3746,32,4205,
     32,4455,32,5001,64,4205,64,5613,32    <016>
4040 DATA 5001,32,4720,64,5001,64,4455,64,
     4205,64,3746,64,3338,32,3746,32,4205  <228>
4050 DATA 32,4455,32,5001,128,0,64,4205,64
     ,5613,64,5001,32,4455,32,4205,64      <181>
4060 DATA 3338,64,3746,32,4205,32,4455,32,
     5001,32,5613,64,5613,64,5001,32,4455  <033>
4070 DATA 32,4205,64,4720,64,5298,64,5613,
     32,5001,32,4720,64,5001,64,5001,32    <149>
4080 DATA 4455,32,4205,32,4455,32,4205,32,
     3746,32,3338,64,5613,64,5001,32       <153>
4090 DATA 4455,32,5001,64,3338,64,0,0      <169>
4999 I=52224                               <217>
5000 FOR C=0 TO 125 STEP 2:READ Q:READ Q1:
     POKE I+2,Q1:HI=INT(Q/256)             <196>
5001 LO=Q-(HI*256):POKE I,HI:POKE I+1,LO:I
     =I+3:NEXT                             <048>
60000 NEW                                  <191>
60004 REM (C) 1985 STEFFEN GOEBBELS        <082>
60005 REM       ALTE HEERSTR.25            <095>
60006 REM         4179 WEEZE 1             <129>
```

**Listing 2. Das Hauptprogramm von »Spion III«. Bitte unbedingt unter dem Namen »D.2« speichern.**

```
0 POKE 808,225:POKE 657,128:IF FL=9 THEN 1
  1536                                     <198>
1 PRINT"{CLR,WHITE}":POKE 53272,23:POKE 53
  280,0:POKE 53281,0:DIM NA$(55):POKE 5000
  2,0:EN=10                                <180>
2 PRINT"{BLUE,UP,SPACE}SPION III ( D 2 MIS
  SION ) VERSION 1.0":DIM XX$(35):DIM BE$(
  32)                                      <197>
3 PRINT TAB(7)"{YELLOW,3DOWN}DESIGNER: STE
  FFEN GOEBBELS":POKE 54272+24,10:POKE 565
  49,30                                    <245>
4 PRINT TAB(7)"{DOWN,WHITE}(C) S.J.G.-SOFT
   WEEZE 1985":ZE=180:GOSUB 60000          <125>
5 DATA 120,169,13,141,20,3,169,195,141,21,
  3,88,96,174,252,3,232,142,252,3,224      <219>
6 DATA 40,208,5,162,0,142,252,3,169,06,157
  ,0,216,169,1,157,1,216,76,0,196          <130>
7 FOR I=49920 TO 49961:READ Q:POKE I,Q:NEX
  T:SYS 49920                              <102>
8 DIM B(55):DIM C(55):DATA 6,10,10,14,10,1
  0,12,7,10,10,11,10,12,5,7,10             <169>
9 DATA 14,10,10,9,5,5,2,11,14,10,14,9,3,10
  ,10,11,10,9:FOR I=0 TO 33:READ B(I):NEXT <024>
10 REM **** DATEN DER LANDSCHAFT ****      <189>
11 DATA 85,247,247,85,223,223,85,247,247,8
   5,223,223,85,247,247,85,223,223,85,247  <132>
12 DATA 247,85,223,223,60,125,60,60,60,60,
   60,60,60,60,60,60,60,60,125,60          <182>
13 DATA 195,215,215,215,215,215,215,215,21
   5,215,215,215,215,215,215,215,215,215   <100>
14 DATA 215,215,215,215,215,20,85,85,85,84
   ,84,84,84,80,80,80,80,64,64,64,64,64    <129>
15 DATA 85,85,85,21,21,21,21,5,5,5,5,1,1,1
   ,1,1,85,106,106,111,111,111,111,111     <191>
16 DATA 85,170,170,255,255,255,255,255,85,
   169,169,249,249,249,249,249,111,111,111 <134>
17 DATA 111,109,109,111,111,255,255,255,25
   5,127,255,255,255,249,249,249,249,249,2
   49                                      <004>
18 DATA 249,249,111,111,111,111,111,111,11
   1,111,255,255,255,255,255,255,255,255   <121>
19 DATA 170,170,191,191,191,191,170,170,17
   0,170,190,190,190,190,170,170,170,170   <137>
20 DATA 254,254,254,254,170,170,191,191,19
   1,191,170,255,85,85,190,190,190,190     <092>
21 DATA 170,255,85,85,254,254,254,254,170,
   255,85,85,170,170,191,191,191,191,191   <031>
22 DATA 191,170,170,255,255,255,255,255,25
   5,170,170,254,254,254,254,254,254       <198>
23 DATA 191,191,191,191,191,191,191,191,25
   4,254,254,254,254,254,254,254           <161>
24 DATA 191,191,191,189,181,149,170,170,25
   5,255,255,85,85,85,170,170,0,0,0,0,0,0  <018>
25 DATA 0,0,254,254,254,126,94,86,170,170,
   85,85,85,85,85,85,85,85                 <160>
26 DATA 255,85,85,85,255,255,255,85,87,87,
   93,93,117,117,213,213,213,213,117,117   <157>
27 DATA 93,93,87,87,85,85,85,86,90,106,85,
   85,85,85,170,170,170,170,85,85,85,85    <089>
28 DATA 170,169,165,149,85,85,85,85,170,10
   6,90,86,85,85,85,85,85,149,165,169,85   <039>
29 DATA 85,85,85,85,85,87,93,117,213,87,93
   ,117,213,85,85,85,85                    <202>
30 DATA 85,85,85,85,213,117,93,87,213,117,
   93,87,85,85,85,85                       <115>
31 DATA 24,24,24,126,126,24,24,24,0,0,1,5,
   5,21,21,21,0,85,85,85,85,85,85,85,0     <008>
32 DATA 192,112,92,95,87,87,87,21,21,21,21
   ,21,21,21,21,87,87,87,87,87,87,87,87    <015>
33 DATA 0,3,5,21,21,85,85,85,0,192,112,92,
   95,87,87,87,87,87,87,87,87,87,84        <252>
34 DATA 0,0,1,255,254,63,31,15,0,0,216,240
   ,99,247,255,222,15,127,30,56,16,128     <155>
35 DATA 128,0,0,0,7,13,36,33,55,15,0,0,128
   ,255,255,255,255,255,15,31,31,27,11,11  <126>
36 DATA 3,1,206,252,248,240,176,0,0,192,31
   ,50,12,63,127,127,255,255,143,191,31    <106>
37 DATA 30,254,231,243,249,64,62,31,63,63,
   31,31,15,0,0,128,192,240,248,248,240    <171>
38 DATA 127,31,3,3,3,1,1,1,252,254,254,252
   ,252,248,248,248,7,7,7,7,3,3,3,2,224    <062>
39 DATA 224,192,128,128,0,0,0,240,240,224,
   96,0,0,0,0                              <067>
40 DATA 127,159,167,169,170,170,170,170,25
   5,255,255,255,127,159,167,169,255,255   <007>
41 DATA 255,255,253,246,218,106,253,246,21
```

**Listing 2. Hauptprogramm von »Spion III« (Fortsetzung)**

```
   8,106,170,170,170,170,253,253,246,246   <211>
42 DATA 218,218,106,106,127,127,159,159,16
   7,167,169,169,170,170,170,170,169,167   <036>
43 DATA 159,127,169,167,159,127,255,255,25
   5,255,106,218,246,253,255,255,255,255   <067>
44 DATA 170,170,170,170,106,218,246,253,10
   6,106,218,218,246,246,253,253,169,169   <104>
45 DATA 167,167,159,159,127,127,170,170,17
   0,170,170,170,170,170                   <134>
46 DATA 170,42,2,0,0,0,0,0,170,170,170,170
   ,10,2,0,0,170,170,170,170,170,170,234   <073>
47 DATA 250,170,170,170,170,170,170,171,17
   5,170,170,170,170,160,128,0,0,170,171   <138>
48 DATA 128,0,0,0,0,0,0,0,0,3,3,15,63,5,
   53,245,245,245,245,245,245,80,85,85     <227>
49 DATA 85,85,84,84,84,63,255,255,51,3,0,0
   ,0,245,250,234,162,162,160,128,0,84     <219>
50 DATA 160,160,128,0,0,0,0,170,170,186,18
   7,171,167,173,191,170,170,174,238,234   <163>
51 DATA 250,250,254,170,170,253,247,223,22
   3,223,117,191,157,183,189,189,189,183   <204>
52 DATA 149,127,253,247,227,119,247,253,24
   7,126,222,246,254,222,118,254,254       <012>
60 REM **** SPRITES ****                   <123>
61 DATA 0,0,0,0,2,170,0,10,170,0,42,170,2,
   170,170,10,170,170,170,170,170,255      <128>
62 DATA 255,255,255,255,255,234,213,83,234
   ,213,67,255,213,3,234,212,3,234,212,3   <097>
63 DATA 255,208,3,255,192,3,234,0,0,212,0,
   0,212,0,0,208,0,0,192,0,0,0:REM SP.0    <067>
64 DATA 0,0,0,170,160,0,170,160,0,170,144,
   0,170,144,0,170,80,0,170,80,0,255,80,0  <139>
65 DATA 255,80,0,171,80,0,171,80,0,255,80,
   0,171,80,0,171,80,0,255,80,0,255,64,0   <254>
66 DATA 3,64,0,3,64,0,3,64,0,3,64,0,3,0,0,
   0:REM SP.1                              <176>
67 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
   0,0,0,0,0,0,1,84,0,5,85,0,26,165,64     <098>
68 DATA 42,169,80,170,170,80,170,170,80,17
   0,170;80,170,170,80,170,171,80,170,170  <248>
69 DATA 80,170,170,80,170,170,80,170,170,8
   0,0:REM SP.2                            <130>
70 DATA 170,170,80,170,170,80,170,170,80,1
   75,170,80,175,170,80,191,234,80,186     <042>
71 DATA 234,80,186,234,80,191,234,80,175,1
   70,80,175,170,80,170,170,80,170,170,80  <061>
72 DATA 170,170,80,170,171,80,170,170,80,1
   70,170,80,170,170,80,170,170,64,42,169  <001>
73 DATA 64,10,165,0,0:REM SP.3             <158>
74 DATA 0,0,0,0,0,0,0,0,149,80,0,169,85,
   0,171,149,80,171,169,85,171,171,255     <155>
75 DATA 171,171,255,171,171,255,171,171,25
   5,171,171,255,171,171,255,43,171,255    <067>
76 DATA 11,171,255,3,171,255,3,171,255,0,1
   71,255,0,43,255,0,11,255,0,3,255,0      <032>
77 DATA 0,0,0,0,8,0,0,42,0,0,46,0,0,46,0,0
   ,42,0,0,8,0,0,42,0,0,42,0,0,170,0,0     <043>
78 DATA 166,0,0,166,0,1,86,0,1,86,0,0,170,
   0,0,170,0,0,170,0,0,170,0,0,42,0,0,12,0 <213>
79 DATA 0,12,0,0:REM SP5                   <145>
80 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
   0,0,0,0,0,42,170,0,90,170,0,90,170      <247>
81 DATA 2,154,170,10,170,170,42,170,170,17
   1,234,170,255,239,255,3,255,255,0,0,0   <179>
82 DATA 0,0,0,0,0,0,0,0,0,0,0,0:REM SP6    <159>
83 DATA 0,40,0,0,40,0,0,168,0,0,168,0,2,16
   0,0,2,160,0,10,160,0,170,172,0,170,188  <020>
84 DATA 0,170,172,0,170,160,0,170,128,0,17
   0,140,0,170,188,0,234,140,0,255,0,0     <179>
85 DATA 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0:RE
   M SP.7                                  <028>
99 REM ****************************        <090>
100 FOR I=2048 TO 2303:READ Q:POKE I,Q:NEX
    T:FOR I=2304 TO 2311:POKE I,0:NEXT     <125>
101 FOR I=2312 TO 2887:READ Q:POKE I,Q:NEX
    T                                      <112>
110 POKE 53270,PEEK(53270)OR 16:POKE 53282
    ,12:POKE 53283,1                       <074>
120 FOR I=0 TO 511:READ Q:POKE I+3072,Q:NE
    XT:VI=53248:V=VI:GOSUB 9000            <046>
130 POKE 2040,48:POKE 2041,49:POKE VI,120:
    POKE VI+2,144:POKE VI+1,110:POKE VI+3,
    110                                    <109>
131 POKE VI+28,255:POKE VI+37,6:POKE VI+38
    ,11:POKE VI+39,6:POKE VI+40,6          <083>
132 POKE 2042,50:POKE 2043,51:POKE VI+4,16
    0:POKE VI+6,160:POKE VI+5,80:POKE VI+7
    ,101                                   <123>
133 POKE VI+41,14:POKE VI+42,14:POKE 2044,
    52:POKE VI+8,238:POKE VI+9,97:POKE VI+
    43,3                                   <243>
134 POKE 2045,53:POKE VI+10,150:POKE VI+11
    ,110:POKE VI+44,6:POKE 2046,54         <183>
135 POKE 2047,55:POKE VI+45,1:POKE VI+46,1
    :GOTO 146                              <204>
136 POKE VI+13,110:POKE VI+15,110:POKE VI+
    21,192:I=370                           <152>
137 I=I-1:F=INT(I/256):G=I-(256*F):H=I-23:
    J=INT(H/256)                           <091>
138 K=H-(256*J):POKE VI+12,K:POKE VI+14,G  <033>
139 IF J=1 THEN POKE VI+16,192             <045>
140 IF J=0 AND F=1 THEN POKE VI+16,128     <239>
141 IF J=0 AND F=0 THEN POKE VI+16,0       <214>
142 IF I<24 THEN F=0:G=0:H=0:J=0:K=0:GOTO
    144                                    <029>
143 GOTO 137                               <087>
144 REM                                    <206>
145 FOR I=23 TO 0 STEP-1:POKE VI+14,I:FOR
    E=0 TO 20:NEXT E,I:RETURN              <237>
146 POKE VI+21,0:POKE VI+12,36:POKE VI+14,
    60:POKE VI+13,152:POKE VI+15,152       <180>
147 POKE VI+16,192                         <246>
148 REM                                    <210>
200 A$="{LIG.RED}@@@@@@@@@@@@@@@@@@@@@@@@@
    @@@@@@@@@@@@@@@":POKE VI+37,11:POKE VI
    +38,12                                 <250>
210 B$="AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
    AAAAAA"                                <233>
220 C$="BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
    BBBBBB":A$=A$+B$+C$+A$+B$+C$           <201>
230 B$="{BROWN}CCCC{ORANGE}@@{BROWN}CCCC{O
    RANGE}@@{BROWN}CCCC{ORANGE}@@{BROWN}CC
    CC{ORANGE}@@{BROWN}CCCC{ORANGE}@@{BROW
    N}CCCC{ORANGE}@@{BROWN}CCCC"           <036>
240 C$="DDDD{ORANGE}AA{BROWN}DDDD{ORANGE}A
    A{BROWN}DDDD{ORANGE}AA{BROWN}DDDD{ORAN
    GE}AA{BROWN}DDDD{ORANGE}AA{BROWN}DDDD{
    ORANGE}AA{BROWN}DDDD":B$=B$+C$         <255>
250 C$="{GREY 3}EEEEEEEEEEEEEEEEEEEEEEEEEE
    EEEEEEEEEEEEEE"                        <127>
260 D$="FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF
    FFFFFF"                                <032>
270 E$="GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG
    GGGGGG":C$=C$+D$+D$+D$+D$+E$           <201>
280 D$="LMN{DOWN,3LEFT}OPQ{DOWN,3LEFT}RSQ" <102>
290 E$="{ORANGE}TUV{DOWN,3LEFT}WXY"        <174>
300 F$="{ORANGE}Z[[[[[[[[[£{DOWN,11LEFT}]S
    SSSSSSSS↑{DOWN,11LEFT}←!!!!!!!!!#"     <191>
310 G$="{ORANGE}Z[[[[[£{DOWN,7LEFT}]SSSSS↑
    {DOWN,7LEFT}←!!!!!#"                   <246>
320 H$="{ORANGE}Z[[£{DOWN,4LEFT}]SS↑{DOWN,
    4LEFT}]SS↑"                            <237>
330 I$="{UP,2RIGHT}"+E$+"{UP,2RIGHT}"+E$+"
    {UP,2RIGHT}"+E$+"{UP,2RIGHT}"+E$+"{UP,
    2RIGHT}"+E$+"{UP,2RIGHT}"+E$+"{UP,2RIG
    HT}"+E$                                <204>
340 J$="{GREY 1}$S{DOWN,2LEFT}$S{DOWN,2LEF
    T}$S{DOWN,2LEFT}$S{DOWN,2LEFT}H{SPACE,
    DOWN,2LEFT}I ":K$="{GREY 1}S${DOWN,2LE
    FT}S${DOWN,2LEFT}S${DOWN,2LEFT}S${DOWN
    ,2LEFT,SPACE}J{DOWN,2LEFT,SPACE}K"     <141>
350 L$="{HOME,6DOWN,ORANGE}$-.$$-.$$-.$$-.
    $$-.$$0/$$0/$$0/$$0/$$0/$"             <247>
360 M$="%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %%%%%%"                                <168>
370 N$="$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
    $$$$$$"                                <187>
371 O$="$$$())))*$$$())))*$$$+)))),$$$+)))
    ),$$$+"                                <161>
380 L$=L$+M$+N$+N$+O$                      <167>
390 M$="SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
    SSSSSS":L$=L$+"{ORANGE}"+M$            <074>
400 N$="{HOME}$$${DOWN,3LEFT}$$${DOWN,3LEF
    T}$$${DOWN,3LEFT}$$${DOWN,3LEFT}$$${DO
    WN,3LEFT}$$${DOWN,3LEFT,DOWN}$"        <098>
401 O$="{HOME,37RIGHT}$$${DOWN,3LEFT}$$${D
    OWN,3LEFT}$$${DOWN,3LEFT}$$${DOWN,3LEF
    T}$$$"                                 <177>
402 P$="{DOWN,3LEFT}$$${2DOWN,LEFT}$":O$=O
    $+P$                                   <151>
403 W$="{40SPACE}"                         <010>
404 T1$="{HOME,5DOWN,8RIGHT,BROWN}HXY{DOWN
```

```
,3LEFT,2SPACE}S{DOWN,3LEFT,2SPACE}S{DO
WN,3LEFT,3SPACE,DOWN,3LEFT,SPACE}"        <113>
405 T2$="{HOME,5DOWN,29RIGHT,BROWN}Z±={DOW
N,3LEFT}S{2SPACE,DOWN,3LEFT}S{2SPACE,D
OWN,3LEFT,3SPACE,DOWN,3LEFT,3SPACE}"      <157>
406 T3$="{HOME,5DOWN,18RIGHT,LIG.GREEN}Z[[
£{DOWN,4LEFT}]{2SPACE}↑"                  <034>
407 T4$="{HOME,6DOWN,ORANGE,10RIGHT}-±※{DO
WN,3LEFT,SHIFT-SPACE}TT"                  <156>
408 T5$="{HOME,14RIGHT,5DOWN,ORANGE}TGG{DO
WN,3LEFT}THZ"                             <232>
410 REM *** NACHRICHTEN : ***             <252>
411 NA$(0)="NORDWALL...KEINE BESONDERHEITE
N."                                       <253>
412 NA$(1)="STADTPARK.EIN PLAETZCHEN ZUM V
ERWEILEN.":NA$(2)=NA$(1):NA$(8)=NA$(1)    <065>
413 NA$(9)=NA$(1):NA$(3)=NA$(0):NA$(4)="DE
R FRIEDHOF...WIE GRUSELIG."               <173>
414 NA$(5)=NA$(0):NA$(6)="ENDE DES NORDWAL
LS.ANFANG EINER PASSAGE":C(6)=8           <107>
415 NA$(7)="DER WESTWALL-DIE HAUPTSTRASSE
DER STADT":NA$(14)=NA$(7)                 <195>
416 NA$(21)=NA$(7):NA$(10)="DIE RINGSTRASS
E...FREI FUER ANLIEGER."                  <069>
417 NA$(12)=NA$(10):NA$(13)="EINE FUSSGAEN
GERPASSAGE..."                            <149>
418 NA$(15)=NA$(10):NA$(16)="TORWEG ZUR PA
RALLELSTRASSE.":C(34)=16                  <161>
419 NA$(17)=NA$(10):NA$(18)=NA$(10):NA$(19
)=NA$(10):NA$(20)=NA$(13)                 <025>
420 NA$(22)="DAS ENDE DER SCHILLERSTRASSE.
"                                         <080>
421 NA$(23)=NA$(16):NA$(24)="DIE SCHILLERS
TRASSE..."                                <080>
422 NA$(25)=NA$(24):NA$(26)=NA$(24):NA$(27
)="ANFANG EINER FUSSGAENGERPASSAGE"       <203>
423 NA$(28)="WESTLICHE INDUSTRIESTRASSE":N
A$(29)="INDUSTRIESTRASSE"                 <082>
424 NA$(31)=NA$(29):NA$(32)="OESTLICHE IND
USTRIESTRASSE."                           <181>
425 NA$(33)="DAS ENDE DER INDUSTRIESTRASSE
."                                        <124>
426 NA$(34)="FRIEDHOF...GRABSTEINE MIT INS
CHRIFTEN."                                <216>
427 NA$(0)="EIN LEBENSMITTELGESCHAEFT(LADE
N)":NA$(35)="IM LEBENSMITTELGESCHAEFT"    <073>
428 C(35)=4:NA$(11)="MEINE WOHNUNG.":NA$(3
6)="IM INNEREN..."                        <183>
429 NA$(30)="TOR B DER CHEMIEWERKE (FABRIK
)":C(36)=96                               <058>
430 NA$(37)="EINE LEERE LAGERHALLE.":B(37)
=6:C(38)=2                                <111>
431 NA$(38)=NA$(37):NA$(43)=NA$(38):NA$(44
)=NA$(38):NA$(45)=NA$(38):C(43)=128       <055>
432 B(38)=8:B(43)=2:B(44)=11:B(45)=8      <015>
433 NA$(37)="AUSGANG : TOR B (SAFE.)":NA$(
38)="AUSGANG : TOR B"                     <110>
434 NA$(42)="EIN FEUCHTER,DUNKLER TUNNEL."
:B(42)=36                                 <147>
435 NA$(39)="EIN BOGENGANG...":NA$(40)=NA$
(39):NA$(41)=NA$(39):NA$(46)=NA$(39)      <115>
436 NA$(47)="DAS KONTROLLZENTRUM.(SCHALTPU
LT,KNOPF)"                                <132>
437 NA$(48)=NA$(39):NA$(53)=NA$(39):NA$(55
)=NA$(39)                                 <046>
438 NA$(54)="EINE GEHEIMNISSVOLLE TUER..."
:B(39)=6:B(40)=10:B(41)=12:B(46)=5        <077>
439 B(47)=4:B(48)=5:B(53)=3:B(54)=10:B(55)
=9:NA$(39)=NA$(39)+" (BLAUER KNOPF)"      <115>
440 TN$="{HOME,4DOWN,3RIGHT,ORANGE}SSSSSSS
SSS{DOWN,10LEFT,LIG.RED}SSSSSSSSSS{DOW
N,10LEFT,GREY 3}SSSSSSSSSS"               <160>
441 TP$="{HOME,WHITE,21DOWN}"             <225>
442 TQ$="{HOME,2RIGHT,BROWN}S{DOWN,LEFT}S{
DOWN,LEFT}S{DOWN,LEFT}S{DOWN,LEFT}S{DO
WN,LEFT}S{DOWN,LEFT}S{DOWN,LEFT}S{DOWN
,LEFT}S{DOWN,LEFT}S{DOWN,LEFT}S":TN$=T
N$+TQ$+TP$:TQ$=""                         <214>
450 PRINT TAB(30)"{WHITE}RETURN...":POKE 1
98,0:WAIT 198,1:POKE 198,0                <031>
460 PRINT"{CLR,DOWN,WHITE}MOECHTEN SIE EIN
{SPACE,CYAN,RVSON}A{RVOFF,WHITE}LTES S
PIEL FORT-"                               <015>
461 PRINT"SETZEN, ODER EIN{SPACE,CYAN,RVSO
N}N{RVOFF,WHITE}EUES BEGINNEN ?":A=0      <221>
462 GET X$:IF X$<>"A"AND X$<>"N"THEN 462  <167>
463 IF X$="A"THEN GOTO 11535              <192>
500 PRINT"{CLR}":POKE 631,13:POKE 198,1   <205>
510 SYS 49152:POKE 50000,B(A):POKE 50001,C
(A):GOSUB 1000:PRINT"{5DOWN}":GOTO 100
00                                        <233>
1000 REM BILD1                            <004>
1001 PRINT"{HOME}"A$                      <061>
1002 PRINT"{HOME}"I$:PRINT"{DOWN,2RIGHT,GR
EY 2}"D$"{2UP,3RIGHT}"E$"{UP,5RIGHT}"
F$"{2UP,RIGHT}"H$"{2UP,RIGHT}"G$;L$       <132>
1003 RETURN                               <043>
1004 REM BILD2                            <137>
1005 PRINT"{HOME}"A$;"{HOME}"I$;"{HOME,4DO
WN}"I$:PRINT"{2UP,LIG.GREEN,UP,2RIGHT
}"D$"{2UP,LIG.BLUE,7RIGHT}"D$;L$          <192>
1006 RETURN                               <046>
1007 REM BILD3                            <012>
1008 PRINT"{HOME}"A$"{HOME}":PRINT TAB(10)
"{3UP}"J$:PRINT"{HOME,2RIGHT}"E$"{HOM
E,2RIGHT,3DOWN}"D$                        <086>
1009 PRINT TAB(15)"{3UP}"F$"{2UP,RIGHT}"F$
"{2UP,RIGHT}Z{DOWN,LEFT}]{DOWN,LEFT}←
"                                         <056>
1010 PRINT L$:POKE 828,15                 <120>
1011 RETURN                               <051>
1012 REM BILD4                            <146>
1013 PRINT"{HOME,GREY 1}";M$;M$;M$;W$;"{WH
ITE}"B$;L$:POKE 828,13                    <005>
1015 RETURN:REM BILD5:                    <086>
1016 PRINT"{HOME,LIG.BLUE}";C$;L$:PRINT"{H
OME,3DOWN,RIGHT,LIG.GREEN,3RIGHT}"H$:
POKE 828,0:RETURN                         <057>
1017 REM BILD 6                           <152>
1018 PRINT"{HOME}"A$;L$;N$                <242>
1019 PRINT"{HOME,UP,4RIGHT}"E$"{UP,8RIGHT}
"E$"{UP,3RIGHT}"E$"{UP,7RIGHT}"E$         <187>
1020 PRINT"{DOWN,4RIGHT,GREY 3}"D$"{2UP,8R
IGHT}"E$"{UP,2RIGHT,LIG.GREEN,RIGHT}"
D$"{2UP,7RIGHT}"E$:RETURN                 <198>
1021 REM BILD 7                           <028>
1022 PRINT"{HOME}"A$;L$;O$;"{HOME}"I$;"{HO
ME,4DOWN}"I$:RETURN                       <058>
1023 REM BILD 8                           <159>
1024 PRINT"{HOME,LIG.BLUE}"C$;L$;"{HOME,LI
G.BLUE,3RIGHT}@@@@@{DOWN,5LEFT}AAAAA{
DOWN,5LEFT}BBBBB{DOWN,5LEFT}@@@@@{DOW
N,5LEFT}AAAAA{DOWN}";                     <224>
1025 PRINT"{5LEFT}BBBBB";N$:RETURN        <000>
1026 REM BILD 9                           <036>
1027 PRINT"{HOME}"A$;L$:PRINT"{HOME,32RIGH
T}"K$                                     <194>
1028 PRINT"{HOME,DOWN,RIGHT,DOWN,GREY 3,DO
WN}"D$"{3RIGHT,2UP}"E$"{3RIGHT,UP}"G$
"{2UP,RIGHT}"H$:RETURN:                   <192>
1029 REM BILD 10                          <083>
1030 PRINT"{HOME}"A$;"{LIG.GREEN}";       <005>
1040 PRINT"{BROWN}$$$$$$-.$$$$$$&$$$$$$$$$
$'$$$$$$0/$$$$$$";                        <236>
1050 PRINT"$$$$-.$$$$$$$&$$$$$$$$$$$$'$$$$
$$$0/$$$$";                               <129>
1060 PRINT"$$-.$$$$$$$$&$$$$$$$$$$$$$$'$$$
$$$$$0/$$";                               <014>
1070 PRINT"-.$$$$$$$$$&$$$$$$$$$$$$$$$$'$$
$$$$$$$0/";                               <137>
1080 PRINT"$$$$$$$$$$&$$$$$$$$$$$$$$$$$$'$
$$$$$$$$$":RETURN                         <063>
1081 REM BILD11 (FRIEDHOF)                <115>
1090 GOSUB 1013:PRINT"{HOME}":PRINT TAB(17
)"{DOWN}1{4SPACE}1{DOWN,5LEFT,4SPACE,
DOWN,4LEFT,4SPACE}":RETURN                <255>
1100 REM BILD 12                          <156>
1110 PRINT"{HOME,GREY 1}"M$;M$;M$;W$;W$;W$
;W$;W$;W$;W$;W$:WW$="{ORANGE}234{DOWN
,3LEFT}5$6{DOWN,3LEFT}5$9"                <108>
1120 WX$="{ORANGE}78{DOWN,2LEFT}$9":PRINT"
{HOME,4DOWN}"WX$"{DOWN}"WW$"{UP}"WW$"
{2UP}"WX$"{4RIGHT,2UP}"WW$"{UP,2RIGHT
}"WW$;                                    <065>
1130 PRINT"{2UP,3RIGHT}"WW$"{4UP,RIGHT}"WX
$"{DOWN,2RIGHT,DOWN}"WW$"{UP,6LEFT,2U
P}"WX$"{RIGHT,UP,LEFT,2UP,LEFT,RIGHT}
"WW$                                      <169>
1135 IF PEEK(50000)AND 16 THEN PRINT T4$  <146>
1140 POKE 828,13:RETURN                   <073>
1200 REM BILD12                           <000>
1210 PRINT"{HOME,BROWN}JKSSSSSSSSSSSSSSSS
SSSSSSSSSSSSSSSSSSLM";                    <045>
1220 PRINT"VVJKSSSSSSSSSSSSSSSSSSSSSSSSSSS
```

**Listing 2. Das Hauptprogramm von »Spion III« (Fortsetzung)**

```
     SSSSSLMVV";                               <018>
1230 PRINT"VVVVJKSSSSSSSSSSSSSSSSSSSSSSSSSS
     SSSLMVVVV";                               <197>
1240 PRINT"VVVVVVJKSSSSSSSSN{CYAN,SPACE}:;
     <=>{BROWN}OSSSSSSSSLMVVVVVV";             <172>
1250 PRINT"VVVVVVVV{LIG.RED}@@@@@@@@V{CYAN
     ,SPACE}?* AB{LIG.RED}V@@@@@@@@VVVVVVV
     V";                                       <159>
1260 PRINT"VVVVVVVVAAAAAAAAV{CYAN,2SPACE}C
     DEF{LIG.RED}VAAAAAAAAVVVVVVVV";           <250>
1270 PRINT"VVVVVVVVBBBBBBBBV{CYAN,2SPACE}G
     H I{LIG.RED}VBBBBBBBBVVVVVVVV";           <049>
1280 PRINT"VVVVVV{LIG.GREEN}POSSSSSSSST{CY
     AN,6SPACE,LIG.GREEN}USSSSSSSSRSVVVVVV
     ";                                        <165>
1290 PRINT"VVVVPOSSSSSSSSSSSSSSSSSSSSSSSSS
     SSSRSVVVV";                               <153>
1291 PRINT"VVPOSSSSSSSSSSSSSSSSSSSSSSSSSSS
     SSSSSRSVV";                               <242>
1292 PRINT"POSSSSSSSSSSSSSSSSSSSSSSSSSSSSS
     SSSSSSSRS";:POKE 828,6:POKE 53283,15      <080>
1295 POKE VI+21,195:RETURN                     <197>
1300 REM BILD 13                               <103>
1310 PRINT"{HOME,ORANGE}SSSSS{BROWN}JKSSSS
     SSSSSSSSSSSSSSSSSSSSSSLM{ORANGE}SSSSS
     ";                                        <192>
1320 PRINT"{ORANGE}SSSSS{BROWN}VVJKSSSSSSS
     SSSSSSSSSSSSSSSLMVV{ORANGE}SSSSS";        <017>
1330 PRINT"{ORANGE}SSSSS{BROWN}VVVVJKSSSSS
     SSSSSSSSSSSSSLMVVVV{ORANGE}SSSSS";        <137>
1340 PRINT"{ORANGE}SSSSS{BROWN}VVVVVVJKSSS
     SSSSSSSSSSSLMVVVVVV{ORANGE}SSSSS";        <051>
1350 PRINT"SSSSSVVVVVVVV$$$$$$$$$$$$$$$VVVV
     VVVVSSSSS";                               <004>
1360 PRINT"SSSSSVVVVVVVV$$$$$$$$$$$$$$$VVVV
     VVVVSSSSS";                               <014>
1370 PRINT"SSSSSVVVVVVVV$$$$$$$$$$$$$$$VVVV
     VVVVSSSSS";                               <024>
1380 PRINT"SSSSSVVVVVV{LIG.GREEN}POSSSSSSS
     SSSSSSSRSVVVVVV{ORANGE}SSSSS";            <015>
1390 PRINT"SSSSS{LIG.GREEN}VVVVPOSSSSSSSSS
     SSSSSSSSSRSVVVV{ORANGE}SSSSS";            <024>
1391 PRINT"SSSSS{LIG.GREEN}VVPOSSSSSSSSSSS
     SSSSSSSSSSSRSVV{ORANGE}SSSSS";            <216>
1392 PRINT"SSSSS{LIG.GREEN}POSSSSSSSSSSSSS
     SSSSSSSSSSSSSRS{ORANGE}SSSSS";:POKE 8
     28,5:POKE 53283,15                        <209>
1394 IF(PEEK(50000)AND 1)OR A=54 THEN PRIN
     T T3$                                     <118>
1395 IF PEEK(50000)AND 2 THEN PRINT T2$        <223>
1396 IF PEEK(50000)AND 8 THEN PRINT T1$        <063>
1397 IF A=35 THEN PRINT T1$:POKE VI+21,208     <198>
1398 IF A=36 THEN PRINT T1$:POKE VI+21,211     <101>
1400 IF A=39 THEN PRINT T5$                    <162>
1401 IF A=45 THEN POKE VI+21,195               <216>
1402 IF A=37 THEN POKE VI+21,204               <175>
1403 IF A=38 THEN POKE VI+21,208               <249>
1404 IF A=46 THEN POKE VI+21,224               <147>
1410 RETURN                                    <198>
1500 PRINT"{HOME}";:FOR I=0 TO 10:PRINT"{O
     RANGE}$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
     $$$$$$$$$";:NEXT                          <056>
1510 POKE 53282,11:RETURN                      <162>
9000 DATA 120,169,31,141,20,3,169,192,141,
     21,3,173,17,208,41,127,141,17,208,169     <119>
9010 DATA 186,141,18,208,169,129,141,26,20
     8,88,96,173,25,208,141,25,208,48,7,17
     3                                         <091>
9020 DATA 13,220,88,76,0,196,173,84,195,20
     1,6,240,24:REM←LAENGE 1.UNTERPRG.52       <038>
9040 DATA 169,6,141,32,208,141,33,208,141,
     84,195,169,1,141,18,208                   <178>
9041 DATA 169,23,141,24,208,76,188,254         <030>
9090 DATA 169,0,141,32,208,141,84,195,169,
     145,141,18,208                            <096>
9091 DATA 169,19,141,24,208,173,60,3,141,3
     3,208,76,188,254                          <146>
9210 DATA 120,169,49,141,20,3,169,234,141,
     21,3,169,240,141,26,208,88,96             <071>
9220 DATA 162,0,189,208,6,157,168,6,232,22
     4,240,208,245,169,21,133,214,169,0        <075>
9230 DATA 133,211,162,0,169,32,157,112,7,2
     32,224,40,208,248,96                      <207>
9300 FOR I=49152 TO 49273:READ Q:POKE I,Q:
     NEXT                                      <129>
9310 FOR I=49408 TO 49441:READ Q:POKE I,Q:
     NEXT                                      <147>
9400 DATA"SCHLUESSEL","SPRENGSATZ","LEBENS
     MITTEL","ZEITUNG","SCHAUFEL"              <252>
9410 DATA"GELD","TASCHENLAMPE","BATTERIE"      <033>
9420 FOR I=0 TO 7:OB(I)=2↑I:NEXT               <053>
9500 FOR I=0 TO 7:READ OB$(I):NEXT             <175>
9600 DATA"NIMM ","VERLIERE ","INV ","INVEN
     TUR ","GRABE ","BETE ","BETRITT "         <022>
9610 DATA"VERLASSE ","ISS ","OEFFNE ","SPR
     ENGE ","DRUECKE ","LIES ","N ","O "       <064>
9620 DATA"S ","W ","H ","R ","SCHLIESSE ",
     "D 2","D2","B2","B 2","1968","1970"       <036>
9630 DATA"WARTE ","PUNKTE ","ZEIT ","WOHNU
     NG","LADEN","FABRIK","FRIEDHOF"           <084>
9640 FOR I=0 TO 32:READ BE$(I):NEXT:RETURN     <241>
10000 X$="":Y$="":Z$="":SYS 49408:GOSUB 51
     000                                       <104>
10010 SYS 49408:L=LEN(X$):I=0:FL=0             <198>
10020 I=I+1:IF MID$(X$,I,1)=" "THEN Y$=LEF
     T$(X$,I):Z$=MID$(X$,I+1,L):GOTO 1003
     2                                         <145>
10030 IF I<=LEN(X$)THEN 10020                  <120>
10031 Y$=X$+" "                                <253>
10032 L=LEN(Z$):I=0                            <135>
10033 I=I+1:IF MID$(Z$,I,1)=" "THEN PRINT"
     ** NUR 2 WORTE BENUTZEN **":Y$="":FL
     =3                                        <024>
10034 IF I<=LEN(Z$)AND FL=0 THEN 10033         <228>
10035 IF XX$(0)=CHR$(13)THEN FL=3              <097>
10040 IF Y$=BE$(13)AND(PEEK(50000)AND 1)TH
     EN A=A-7:FL=1                             <170>
10041 IF Y$=BE$(14)AND(PEEK(50000)AND 2)TH
     EN A=A+1:FL=1                             <113>
10042 IF Y$=BE$(15)AND(PEEK(50000)AND 4)TH
     EN A=A+7:FL=1                             <127>
10043 IF Y$=BE$(16)AND(PEEK(50000)AND 8)TH
     EN A=A-1:FL=1                             <201>
10044 IF Y$=BE$(18)AND(PEEK(50000)AND 16)T
     HEN A=42:FL=1                             <225>
10045 IF Y$=BE$(17)AND(PEEK(50000)AND 32)T
     HEN A=34:FL=1                             <148>
10046 IF(X$="N"OR X$="S"OR X$="W"OR X$="O"
     )AND FL=0 THEN PRINT"WEG IST VERSPER
     RT":FL=3                                  <138>
10047 IF(X$="H"OR X$="R")AND FL=0 THEN PRI
     NT"WEG IST VERSPERRT":FL=3                <062>
10048 IF A=49 THEN A=39:SYS 49408:PRINT"HI
     NTER MIR FAELLT DER TUNNEL EIN...":S
     YS 49408                                  <250>
10050 REM                                      <206>
10060 IF Y$<>BE$(2)AND Y$<>BE$(3)THEN 1010
     0                                         <207>
10061 PRINT"FOLGENDE DINGE FUEHRE ICH MIT
     MIR:":SYS 49408                           <019>
10062 FOR I=0 TO 7:IF PEEK(50002)AND OB(I)
     THEN PRINT OB$(I):SYS 49408:GOSUB 10
     080                                       <089>
10063 NEXT                                     <167>
10064 IF(PEEK(50002))=0 THEN PRINT"ICH BIN
     ARM WIE EINE KIRCHENMAUS."                <030>
10065 FL=3                                     <171>
10071 GOTO 10090                               <195>
10080 POKE 198,0:WAIT 198,1:POKE 198,0:RET
     URN                                       <234>
10090 FL=3                                     <196>
10100 IF Y$="TOETE "OR Y$="ZERSTOERE "THEN
     FL=9                                      <194>
10101 IF Y$=BE$(5)THEN PRINT"VIELLEICHT WI
     RD DADURCH DAS WETTER":FL=9               <014>
10102 IF FL=9 THEN FL=3:SYS 49408:PRINT"BE
     SSER. ABER ICH GLAUBS' KAUM.":WE=1        <157>
10103 REM                                      <003>
10104 IF FL=9 THEN PRINT"DENKEN SIE AN DAS
     STRAFGESETZBUCH.":FL=3                    <045>
10110 IF Y$<>BE$(6)THEN 10200                  <035>
10111 IF Z$=BE$(32)AND A=4 THEN A=34:FL=1:     <101>
10112 IF Z$=BE$(32)AND FL<>1 THEN PRINT"WO
     SOLL HIER EIN "BE$(32)" SEIN PRINT"
     :FL=1                                     <200>
10120 IF Z$=BE$(30)AND A=0 THEN A=35:FL=1      <169>
10121 IF Z$=BE$(30)AND FL<>1 THEN PRINT"DA
     S IST HIER NICHT.":FL=3                   <127>
10130 IF Z$=BE$(29)AND A=11 THEN A=36:FL=1     <134>
10131 IF Z$=BE$(29)AND FL<>1 THEN PRINT"DA
     S IST HIER NICHT.":FL=1                   <189>
10140 IF Z$=BE$(31)AND(A=29 OR A=30)THEN A
```

```
       =37:FL=1                                <204>
10141 IF Z$=BE$(31)AND FL=0 THEN PRINT"HIE
       R IST KEIN EINGANG...":FL=3             <179>
10199 IF FL=0 THEN PRINT"DAS KANN ICH NICH
       T BETRETEN !":FL=3                      <235>
10200 REM                                      <100>
10210 IF Y$<>BE$(7)THEN 10300                  <144>
10211 IF Z$=BE$(32)AND A=34 THEN A=4:FL=1:     <231>
10212 IF Z$=BE$(32)AND FL<>1 THEN PRINT"WO
       SOLL HIER EIN "BE$(32)" SEIN ?":FL=
       1                                       <245>
10220 IF Z$=BE$(30)AND A=35 THEN A=0:FL=1      <145>
10221 IF Z$=BE$(30)AND FL=0 THEN PRINT"ICH
       BIN NICHT DARIN.":FL=3                  <110>
10230 IF Z$=BE$(29)AND A=36 THEN A=11:FL=1     <117>
10231 IF Z$=BE$(29)AND FL=0 THEN PRINT"ICH
       BIN NICHT DARIN.":FL=3                  <188>
10240 IF Z$=BE$(31)AND(A=37 OR A=38)THEN A
       =29:FL=1                                <097>
10241 IF Z$=BE$(31)AND FL=0 THEN PRINT"WIE
       SOLL ICH DAS VERLASSEN ?":FL=3          <186>
10299 IF FL=0 THEN PRINT"DAS KANN ICH NICH
       T VERLASSEN!":FL=3                      <179>
10300 IF Y$<>BE$(12)THEN 10400                 <033>
10310 IF(Z$="INSCHRIFT"OR Z$="INSCHRIFTEN"
       )AND A=34 THEN FL=9                     <223>
10320 IF FL=9 THEN PRINT"PORTA EST ALIENA
       TERRA...":                              <027>
10330 IF FL=9 THEN SYS 49408:PRINT"ANDRE'
       MOLL":SYS 49408:PRINT"* "BE$(24)" +.
       .":FL=3                                 <176>
10400 I=0:IF Y$<>BE$(0)AND Y$<>"NEHME "THE
       N 10500                                 <016>
10401 IF Z$=OB$(2)AND PEEK(50001)AND 4 AND
       GE=0 THEN PRINT"KEIN GELD...":FL=3:
       GOTO 10500                              <209>
10402 IF Z$=OB$(I)THEN 10450                   <187>
10410 I=I+1:IF I<8 THEN 10401                  <010>
10420 FL=4:GOTO 10470                          <159>
10450 IF PEEK(50001)AND OB(I)THEN C(A)=C(A
       )-OB(I):POKE 50002,PEEK(50002)+OB(I)
       :FL=1                                   <058>
10460 IF FL=0 THEN PRINT"ICH SEHE DAS HIER
       NIRGENDS.":FL=3                         <069>
10470 IF Z$="ALLES"AND A<>35 THEN POKE 500
       02,PEEK(50002)+C(A):C(A)=0:FL=1         <048>
10480 IF Z$="ALLES"AND A=35 THEN PRINT"BIT
       TE GENAUERE EINGABE !":FL=3             <004>
10500 IF Y$<>BE$(1)AND Y$<>"LEGE "THEN 106
       00                                      <089>
10501 I=0                                      <020>
10502 IF Z$=OB$(5)THEN GE=0                    <177>
10510 IF OB$(I)=Z$THEN 10540                   <080>
10520 I=I+1:IF I<8 THEN 10510                  <249>
10530 FL=4:GOTO 10590                          <035>
10540 IF PEEK(50002)AND OB(I)THEN POKE 500
       02,PEEK(50002)-OB(I):C(A)=C(A)+OB(I)
       :FL=1                                   <145>
10550 IF FL=0 THEN PRINT"DAS BESITZE ICH N
       ICHT...":FL=3                           <159>
10590 IF Z$="ALLES"THEN C(A)=C(A)+PEEK(500
       02):POKE 50002,0:FL=1                   <038>
10600 IF Y$=BE$(4)AND A<>34 THEN PRINT"GRA
       BEN SIE AN EINEM ANDEREN ORT!?":FL=3    <097>
10601 IF Y$=BE$(4)AND FL=0 AND PEEK(50002)
       AND 16 THEN PRINT"ICH GRABE EIN LOCH
       .":FL=9                                 <176>
10602 IF Y$=BE$(4)AND FL=0 THEN PRINT"KEIN
       E SCHAUFEL VORHANDEN.":FL=3             <020>
10603 IF FL<>9 THEN 10700                      <234>
10604 B(34)=16:FL=3                            <001>
10700 IF Y$=BE$(28)THEN PRINT"NOCH "ZE" MI
       N. BIS ZUM START.":FL=3                 <071>
10701 IF Y$=BE$(27)THEN PRINT"IHRE PUNKTZA
       HL : "SC:FL=3                           <220>
10702 IF Y$=BE$(8)AND(PEEK(50002)AND 4)THE
       N FL=1:POKE 50002,PEEK(50002)-4:HU=1    <045>
10703 IF Y$=BE$(8)AND FL=0 THEN PRINT"WAS
       SOLL ICH ESSEN ?":FL=3                  <166>
10704 IF Y$="STIRB "THEN POKE 50000,0:SYS
       50000                                   <008>
10705 IF Y$=BE$(8)AND FL=1 AND A=35 THEN P
       RINT"ICH VERSCHMUTZE DEN LADEN":FL=9    <056>
10706 IF FL=9 THEN SYS 49408:PRINT"ICH ERW
       ARTE EIN DISZIPLINARVERFAHREN..":GOT
       O 60110                                 <049>
10710 IF Y$<>"SCHALTE "THEN 10800              <156>
10720 IF Z$<>OB$(6)THEN PRINT"DAS KANN ICH
       NICHT ANSCHALTEN...":FL=3:GOTO 1080
       0                                       <099>
10730 IF(PEEK(50002)AND 64)THEN 10740          <014>
10731 PRINT"ICH HABE KEINE "OB$(6):FL=3:GO
       TO 10800                                <062>
10740 IF(PEEK(50002)AND 128)THEN 10750         <150>
10741 PRINT"ICH BRAUCHE EINE "OB$(7):FL=3:
       GOTO 10800                              <112>
10750 FL=1:LA=1                                <058>
10800 IF A=42 AND LA=0 THEN 10810              <066>
10801 GOTO 10890                               <164>
10810 SYS 49408:PRINT"ICH BETRETE EINEN TU
       NNEL...":FOR I=0 TO 1000:NEXT           <041>
10820 SYS 49408:PRINT"ICH KANN LEIDER WEGE
       N DER DUNKELHEIT":SYS 49408             <034>
10830 PRINT"NICHTS SEHEN....ICH STOSSE MIC
       H UND ":SYS 49408                       <047>
10840 PRINT"VERBLUTE...AGHROOHJAAAH":GOTO 6
       0110                                    <110>
10890 IF HU=1 OR ZE>150 THEN 10900             <141>
10891 SYS 49408:PRINT"ICH HABE SCHON SEIT
       TAGEN NICHTS MEHR ":SYS 49408           <026>
10892 PRINT"GEGESSEN ODER GETRUNKEN. ICH H
       ABE":SYS 49408                          <183>
10893 PRINT"KEINE LUST MIT LEEREM BAUCH ZU
       ARBEITEN":GOTO 60110                    <224>
10900 IF Y$=BE$(26)THEN ZE=ZE-10:FL=1          <108>
10910 IF Y$=BE$(11)AND A<>47 AND A<>39 THE
       N PRINT"ICH SEHE HIER KEINEN KNOPF":
       FL=3                                    <189>
10920 IF Y$=BE$(11)AND A=47 THEN 10930         <134>
10925 IF Y$=BE$(11)AND A=39 THEN 10960         <079>
10926 GOTO 10990                               <065>
10930 SYS 49408:PRINT"ICH SELBST HABE DEN
       ZUENDMECHANISMUS":SYS 49408             <106>
10940 PRINT"IM SPACE SHUTTLE AKTIVIERT(NOH
       ER WISSEN":SYS 49408                    <223>
10950 PRINT"SIE,DASS ICH EIN DOPPELAGENT B
       IN ?":GOTO 60110                        <067>
10960 SYS 49408:PRINT"ICH HOERE IN DER FER
       NE EINE EXPLOSION":SYS 49408:           <226>
10961 PRINT"EIN DUMPFES GROLLEN...ES WIRD
       IMMER":SYS 49408                        <024>
10962 PRINT"LAUTER. ES SCHEINT SO, ALS HAE
       TTE ICH":SYS 49408                      <043>
10963 PRINT"DEN STAUDAMM GESPRAENGT.....GL
       UCKBLUBB":GOTO 60110                    <191>
10990 IF Y$="LIES "AND(PEEK(50002)AND 8)AN
       D FL=0 THEN 11000                       <032>
10991 IF Y$=BE$(12)AND FL=0 THEN PRINT"WAS
       SOLL ICH LESEN ?":FL=3:GOTO 11010       <176>
10992 GOTO 11010                               <112>
11000 PRINT"ICH LESE DIE ZEITUNG :":SYS 49
       408:PRINT"DER FLUG DES SPACE SHUTTLE
       S IST"                                  <198>
11001 SYS 49408:PRINT"VOELLIG (!) PROBLEML
       OS UND ERFOLGREICH":FL=3:POKE 198,0     <247>
11002 WAIT 198,1:SYS 49408:PRINT TAB(17)"-
       -----":SYS 49408:PRINT"HOROSKOPE :"     <067>
11003 SYS 49408:PRINT"[...] UND NEHMEN SIE
       SICH VOR KNOEPFEN":SYS 49408:SC=SC+
       10                                      <228>
11004 PRINT"IN ACHT ! [...]":POKE 198,0        <157>
11010 IF Y$<>BE$(19)THEN 11100                 <088>
11011 IF A<>54 THEN PRINT"FUNKTIONIERT HIE
       R NICHT":FL=3:GOTO 11100                <241>
11020 IF PEEK(50002)AND 1 THEN 11030           <232>
11025 PRINT"ICH HABE KEINEN SCHLUESSEL..."
       :FL=3:GOTO 11100                        <191>
11030 B(54)=11:FL=1:SC=SC+50                   <141>
11100 IF Y$<>BE$(10)THEN 11200                 <036>
11110 IF PEEK(50002)AND 2 THEN 11130           <083>
11120 PRINT"ICH HABE KEINEN SPRENGSATZ.":F
       L=3:GOTO 11200                          <248>
11130 IF A<>47 THEN 11170                      <240>
11131 IF PEEK(50002)AND 8 THEN PRINT"DIE Z
       EITUNG ENTZUENDET SICH...":FL=10        <005>
11132 IF FL=10 THEN SYS 49408:PRINT"ICH VE
       RBRENNE.SCHADE,SCHON TOT...":GOTO 60
       110                                     <064>
11140 PRINT"ICH HABE DEN COUNTDOWN STOPPEN
       KOENNEN":SYS 49408:SC=SC+200            <114>
11150 PRINT"MEINE MISSION IST ERFUELLT.":S
       YS 49408                                <143>
11160 PRINT"IHRE PUNKTE :";SC:SYS 49408:PR
       INT"{CYAN}(TASTENDRUCK)"                <230>
```

**Listing 2. Das Hauptprogramm von »Spion III« (Fortsetzung)**

```
11161 POKE 198,0:WAIT 198,1:GOSUB 50000:PO
      KE 50000,0:SYS 50000                 <099>
11170 PRINT"ICH SPRENGE MICH SELBST...BOUO
      OUMMMM..":GOTO 60110                 <053>
11200 IF ZE<1 THEN SYS 49408:PRINT"ZU SPAE
      T...":GOTO 60110                     <123>
11210 IF Y$<>BE$(9)THEN 11300              <016>
11220 IF A=37 THEN 11230                   <138>
11225 PRINT"ICH KANN HIER NICHTS OEFFNEN."
      :FL=3:GOTO 11300                     <149>
11230 PRINT"ICH VERSUCHE DEN SAFE ZU OEFFN
      EN.":FL=1:C(37)=1                    <241>
11240 SYS 49408:PRINT"ICH BENOETIGE JEDOCH
       EINE ZAHLEN -":SYS 49408:X$=""      <231>
11250 PRINT"KOMBINATION : ";:GOSUB 51001   <067>
11251 IF X1$=BE$(24)THEN SYS 49408:SC=SC+6
      0:GOTO 11300                         <137>
11252 SYS 49408:PRINT"DIE KOMBINATION IST
      LEIDER FALSCH.":SYS 49408            <135>
11260 PRINT"EINE ALARMANLAGE WIRD AKTIVIER
      T.":SYS 49408                        <046>
11270 PRINT"ICH WERDE VERHAFTET....":GOTO
      60110                                <150>
11300 IF A=46 AND HH=1 THEN SYS 49408:PRIN
      T"DER ROBOTER TOETET MICH.ORGH...":G
      OTO 60110                            <136>
11310 IF A=46 AND HH=0 THEN SYS 49408:PRIN
      T"EIN GEFAEHRLICHER ROBOTER..":SYS 4
      9408:HH=1                            <243>
11315 IF A=36 THEN T=T+1                   <038>
11320 IF A=36 AND T=2 THEN 11340           <202>
11330 GOTO 11400                           <006>
11340 SYS 49408:PRINT"DAS TELEFON KLINGELT
      .":SYS 49408                         <116>
11350 PRINT"MEIN CHEF IST AM APPARAT.":SYS
       49408                               <239>
11351 PRINT"ER FRAGT NACH DEM NAMEN MEINER
       MISSION":SYS 49408                  <148>
11352 PRINT"->";:GOSUB 51001:I=19          <068>
11353 I=I+1:IF X1$=BE$(I)THEN SYS 49408:SC
      =SC+110:GOTO 11400                   <163>
11354 IF I<24 THEN 11353                   <198>
11355 SYS 49408:PRINT"ICH HABE MEINEN CHEF
       BELOGEN !":SYS 49408                <192>
11356 PRINT"ICH WERDE BESTRAFT. DAS SIND J
      A MITTEL-":SYS 49408                 <191>
11357 PRINT"ALTERLICHE SITTEN AUUUUUR...":
      GOTO 60110                           <192>
11400 IF Y$="VERDAMMT "OR Y$="SCHEISSE "OR
       Y$="MIST "THEN FL=9                 <042>
11401 IF FL=9 THEN PRINT"FLUCHEN HILFT NIC
      HTS...":FL=3                         <145>
11402 IF X$="HILFE"THEN PRINT"JEDER IST SI
      CH SELBST DER NAECHSTE.":FL=3        <014>
11403 IF WE=0 AND ZE<100 THEN SYS 49408:FL
      =9:PRINT"MEINE SACHEN WERDEN DURCH D
      EN SEHR"                             <032>
11404 IF FL=9 THEN SYS 49408:PRINT"STARKEN
       WIND IN ALLE RICHTUNGEN GEWEHT"     <003>
11405 IF FL<>9 THEN 11500                  <006>
11410 FOR I=0 TO 7                         <047>
11411 : R=INT(32*RND(1))+1                 <171>
11412 : IF PEEK(50002)AND 2↑I THEN C(R)=C(
      R)+2↑I                               <019>
11413 NEXT I:SYS 49408:WE=1:POKE 50002,0:L
      A=0                                  <127>
11500 IF LA=1 THEN EN=EN-1:IF EN=1 THEN LA
      =0:SYS 49408:PRINT"BATTERIEEN SIND L
      EER":FM=9                            <120>
11501 IF FM=9 THEN SYS 49408:FM=0:POKE 500
      02,PEEK(50002)-128                   <094>
11502 IF SC>255 THEN SC=255                <225>
11503 IF SC<1 THEN SC=0                    <252>
11504 IF PEEK(50002)AND 192 THEN 11510     <025>
11505 LA=0                                 <181>
11510 IF Y$<>"SAVE "THEN 11600             <035>
11520 INPUT"FILE-NAMEN ";X$                <230>
11521 FOR I=0 TO 55:POKE 50003+I,C(I):NEXT
      :BA=50059:POKE BA,ZE:POKE BA+1,SC    <175>
11522 POKE BA+2,LA:POKE BA+3,B(34):POKE BA
      +4,B(54):POKE BA+5,T:POKE BA+6,WE    <113>
11523 POKE BA+7,HU:POKE BA+8,HH:POKE BA+9,
      EN:POKE BA+10,A:POKE BA+11,GE        <030>
11524 REM DATEN IM BEREICH 50002-50070     <184>
11530 SYS 49408:PRINT"BITTE DISK EINSCHALT
      EN (RETURN...).":POKE 198,0          <161>
11531 WAIT 198,1:POKE 198,0:FOR I=0 TO 4:S
      YS 49408:NEXT:PRINT"{HOME,17DOWN}"   <244>
11532 GOSUB 11550:OPEN 1,8,1,X$:PRINT#1,CH
      R$(82);CHR$(195);                    <041>
11533 FOR I=50002 TO 50070:W=PEEK(I):PRINT
      #1,CHR$(W);:NEXT:SYS 49408:FL=3:POKE
       198,0                               <032>
11534 CLOSE 1:PRINT"{HOME,22DOWN}":SYS 494
      08:GOTO 11600                        <003>
11535 INPUT"FILE-NAMEN ";X$:FL=9:GOSUB 115
      80:LOAD X$,8,1                       <201>
11536 FOR I=0 TO 55:C(I)=PEEK(50003+I):NEX
      T:BA=50059:ZE=PEEK(BA):SC=PEEK(BA+1) <105>
11538 LA=PEEK(BA+2):B(34)=PEEK(BA+3):B(54)
      =PEEK(BA+4):T=PEEK(BA+5):WE=PEEK(BA+
      6)                                   <019>
11540 HU=PEEK(BA+7):HH=PEEK(BA+8):EN=PEEK(
      BA+9):A=PEEK(BA+10):FL=0             <206>
11541 GE=PEEK(BA+11):GOTO 500              <172>
11550 POKE 631,71:POKE 632,79:POKE 633,84:
      POKE 634,79:POKE 635,49:POKE 636,49  <184>
11551 POKE 637,53:POKE 638,54:POKE 639,48:
      POKE 640,13:POKE 198,10:RETURN       <097>
11560 CLOSE 1:FOR I=0 TO 3:SYS 49408:NEXT:
      PRINT"FILE KONNTE NICHT GE-SAVED WER
      DEN."                                <036>
11570 SYS 49408:GOTO 11520                 <039>
11580 POKE 631,71:POKE 632,79:POKE 633,84:
      POKE 634,79:POKE 635,49:POKE 636,49  <214>
11581 POKE 637,53:POKE 638,56:POKE 639,50:
      POKE 640,13:POKE 198,10:RETURN       <249>
11582 PRINT"{CLR}FILE KONNTE NICHT GELADEN
       WERDEN.":POKE 198,0:WAIT 198,1:POKE
       198,0                               <054>
11583 GOTO 460                             <201>
11600 REM                                  <232>
19000 POKE 50000,B(A):REM RICHTUNGEN       <176>
19010 POKE 50001,C(A):REM GEGENSTAENDE     <059>
19020 POKE 53282,12:POKE VI+21,192         <062>
19030 IF PEEK(50002)AND 32 THEN GE=1       <162>
19997 ZE=ZE-1:SC=SC-2:IF FL=4 THEN PRINT"D
      IESES OBJEKT KENNE ICH NICHT."       <236>
19998 IF FL=1 THEN PRINT"O.K."             <223>
19999 IF FL=0 THEN PRINT"HUGH?":ZE=ZE-1    <130>
20000 IF A>12 THEN 20003                   <059>
20001 ON A+1 GOSUB 1000,1012,1012,1004,109
      0,1000,1004,1004,1012,1012,1000,1004
      ,1004                                <189>
20002 GOTO 20008                           <166>
20003 IF A>24 THEN 20006                   <110>
20004 ON A-13+1 GOSUB 1030,1018,1000,1007,
      1004,1004,1022,1030,1000,1000,1027,1
      004                                  <169>
20005 GOTO 20008                           <169>
20006 IF A>36 THEN 20009                   <065>
20007 ON A-25+1 GOSUB 1000,1004,1022,1024,
      1016,1016,1000,1004,1022,1100,1300,1
      300                                  <006>
20008 GOTO 20030                           <104>
20009 ON A-37+1 GOSUB 1300,1300,1300,1300,
      1300,1500,1300,1300,1300,1300,1200,1
      300                                  <205>
20010 IF A<49 THEN 20030                   <206>
20020 ON A-49+1 GOSUB 1500,1500,1500,1500,
      1300,1300,1300                       <071>
20030 REM                                  <024>
29009 PRINT"{HOME,12DOWN,CYAN}RICHTUNGEN:{
      12SPACE,12LEFT,WHITE}";              <153>
29010 IF PEEK(50000)AND 32 THEN PRINT"H ,"
      ;                                    <201>
29011 IF PEEK(50000)AND 16 THEN PRINT"R ,"
      ;                                    <062>
29012 IF PEEK(50000)AND 1 THEN PRINT"N ,"; <252>
29013 IF PEEK(50000)AND 2 THEN PRINT"O ,"; <014>
29014 IF PEEK(50000)AND 4 THEN PRINT"S ,"; <051>
29015 IF PEEK(50000)AND 8 THEN PRINT"W ,"; <120>
29016 PRINT"{HOME,13DOWN}";W$;             <041>
29017 PRINT W$;W$                          <235>
29018 PRINT"{4UP,CYAN}BESONDERHEITEN:{WHIT
      E,SPACE}";                           <128>
29019 FOR I=0 TO 1:IF PEEK(50001)AND OB(I)
      THEN PRINT OB$(I)",";                <148>
29020 NEXT:PRINT                           <178>
29021 FOR I=2 TO 4:IF PEEK(50001)AND OB(I)
      THEN PRINT OB$(I)",";                <088>
29022 NEXT:PRINT"{HOME,14DOWN}";           <100>
```

```
29023 FOR I=5 TO 7:IF PEEK(50001)AND OB(I)
      THEN PRINT OB$(I)",";                  <059>
29024 NEXT                                   <076>
29092 PRINT"{HOME,16DOWN}"W$:PRINT"{2UP,CY
      AN}"NA$(A)"{WHITE}"                    <236>
29093 PRINT TP$                              <024>
30000 GOTO 10000                             <246>
50000 PRINT"{CLR,WHITE}":SYS 49256:POKE 53
      280,0:POKE 53281,0:POKE 53272,21:POK
      E VI+21,0                              <011>
50001 SYS 50452:FOR I=0 TO 30                <092>
50002 : R=INT(999*RND(1))                    <154>
50003 : POKE 1024+R,46                       <063>
50004 NEXT                                   <229>
50005 GOSUB 136:POKE 198,0:PRINT"{HOME,10D
      OWN,15RIGHT,RED}TSCHUESS..."           <208>
50006 FOR I=0 TO 3000:NEXT:RETURN            <083>
51000 PRINT"{WHITE}>";                       <122>
51001 POKE 204,0:I=0:REM INPUT ROUTINE:      <123>
51002 GET XX$(I):IF XX$(I)=""THEN 51002      <075>
51003 IF(ASC(XX$(I))>31)AND(ASC(XX$(I))<12
      8)THEN 51100                           <183>
51004 IF(ASC(XX$(I))>193)AND(ASC(XX$(I))<2
      19)THEN 51100                          <243>
51005 IF(ASC(XX$(I))=157)OR(ASC(XX$(I))=20
      )THEN 52000                            <061>
51006 IF ASC(XX$(I))=13 THEN 53000           <092>
51007 IF ASC(XX$(I))=255 THEN REM            <172>
51008 GOTO 51002                             <220>
51100 PRINT XX$(I);:I=I+1:IF I>34 THEN I=3
      4:PRINT"{LEFT,SPACE,LEFT}";:           <005>
51101 GOTO 51002                             <057>
52000 IF I>-1 THEN PRINT"{SPACE,2LEFT,SPAC
      E,LEFT}";:I=I-1:IF I<0 THEN PRINT">"
      ;:I=0                                  <211>
52001 GOTO 51002                             <197>
53000 X1$="":POKE 207,0:POKE 204,1:FOR AA=
      0 TO I-1:X1$=X1$+XX$(AA):NEXT:PRINT"
       "                                     <080>
53002 X$=X1$:RETURN                          <022>
60000 PRINT"{WHITE,3DOWN}BITTE WARTEN SIE.
      ..":PRINT"{DOWN,CYAN}DIE BEFEHLSLIST
      E :"                                   <176>
60001 PRINT"{WHITE}INV (INVENTUR),BETRITT,
      VERLASSE,LIES{4SPACE}";                <127>
60002 PRINT"{WHITE}N,O,S,W,H,R,NIMM (NEHME
      ),VERLIERE (LEGE)";                    <043>
60003 PRINT"{WHITE}GRABE,ZEIT,PUNKTE,ISS,S
      TIRB,SCHALTE,BETE";                    <102>
60004 PRINT"{WHITE}WARTE,DRUECKE,SCHLIESSE
      ,SPRENGE,OEFFNE{2SPACE}";              <180>
60005 PRINT"SAVE...UND EINIGE MEHR..."       <195>
60006 PRINT"{DOWN}MUSIK: NUN RUHEN ALLE WA
      ELDER (J.S.BACH)"                      <195>
60007 PRINT"{WHITE}THANKS TO PEANUTS SOFT.
      ...{14SPACE}";                         <033>
60008 PRINT"GRUESSE AN ALLE SCHILDKROETEN
      & STUEMPER";                           <028>
60100 PRINT"{WHITE}VIEL GLUECK....";:RETUR
      N                                      <121>
60110 GOSUB 1100:POKE VI+21,0:PRINT TN$:SY
      S 49408:PRINT"NOCH EIN VERSUCH (J/N)
       ";                                    <028>
60115 POKE 204,0:POKE 207,0:GET X$:IF X$="
      "THEN 60115                            <177>
60120 IF X$<>"J"AND X$<>"N"THEN 60110        <131>
60130 IF X$="J"THEN POKE 204,1:POKE 207,0:
      SYS 49256:RUN                          <178>
60140 SYS 49408:PRINT"THANKS FOR PLAYING."
      :FOR I=0 TO 2000:NEXT:POKE 50000,0:S
      YS 50000                               <090>
60150 REM SPION III                          <203>
60151 REM VON STEFFEN GOEBBELS               <062>
60152 REM ALTE HEERSTR.25                    <242>
60153 REM 4179 WEEZE 1                       <020>
60154 REM (C) 1985 S.J.G.-SOFT               <199>
```

Listing 2. Das Hauptprogramm von »Spion III« (Schluß)

# The Sword – die Magierprüfung

**Entfliehen Sie ins Land der Träume. Haben Sie teil am Treiben der Zauberer und Dämonen. Bestehen Sie die Aufnahmeprüfung des dunklen Zirkels.**

Machen Sie es sich zur Aufgabe, von dem kleinen Zauberlehrling, der Sie gerade sind, zum Meister der Schwarzen Magie aufzusteigen. Absolvieren Sie den letzten Test der hohen Mächte. »The Sword« ist ein deutsches Textadventure, das ein wenig nach dem Vorbild der Infocom-Adventures gestaltet wurde. Der Sprachschatz enthält über 100 Wörter, die auch als ganze Sätze eingegeben werden können. Zur Auflockerung kann zwischen dem normalen und einem geänderten Zeichensatz gewählt werden.

Neben den allgemein üblichen Befehlen sind allerdings auch noch ein paar unübliche enthalten. So zum Beispiel:

»FARBE« – Damit läßt sich die Schrift- und Bildschirmfarbe Ihren Wünschen gemäß einstellen.

»NORMAL« – Schaltet von dem neuen auf den Commodore-Zeichensatz zurück.

»NEU« – Dies ist das Gegenstück zu »NORMAL«.

»HILFE« – Dieser Befehl sollte nur eingegeben werden, wenn man kurz vor dem Verzweifeln ist.

»ANLEITUNG« – Hier erhalten Sie einige Tips zum Spiel.

»SAVE«, »STORE«, »SPEICHERN« – Mit diesen Befehlen kann man drei verschiedene Spielstände speichern.

»LOAD«, »LADE«, »RESTORE« – Diese Befehle laden einen alten Spielstand.

Bitte geben Sie zuerst die Programme »A« (Listing 1) und »B« (Listing 2) mit Hilfe des MSE ein (zu finden in diesem Heft), und speichern Sie diese auf Ihrem Datenträger. Anschließend tippen Sie bitte das Hauptprogramm (Listing 3) ein, speichern es ebenfalls und starten es mit »RUN«. Auf dem Bildschirm erscheint dann die Meldung »LOADING...«, und der neue Zeichensatz sowie die Maschinenroutinen werden nachgeladen. Sodann erscheint das Titelbild, und die Daten werden eingelesen. Hier werden Sie nun gefragt, ob Sie mit dem normalen oder dem neu definierten Zeichensatz spielen wollen. Geben Sie bitte nur »J« oder »N« ein. Ist dies geschehen, befinden Sie sich schon mitten im Spiel.

(A. Sommer/S. Adomat/dm)

**Listing 1. »A«**
**(bitte mit dem MSE eingeben)**

```
 programm : a                    c800 d000
-------------------------------------------
c800 : 3c 42 99 a1 99 42 3c 00   94
c808 : 00 40 3c 02 3e 42 3e 00   67
c810 : 20 40 40 7c 42 22 1c 00   96
c818 : 00 00 3c 42 40 40 3c 02   6b
c820 : 04 02 02 3e 42 42 3c 00   95
c828 : 00 00 3c 42 7e 40 3c 02   5f
c830 : 01 0e 10 3e 08 08 08 10   05
c838 : 00 00 38 44 44 3e 02 7c   06
c840 : 20 40 40 7c 42 42 24 00   e7
c848 : 04 18 00 38 08 08 0e 00   58
c850 : 01 06 00 04 02 02 04 38   86
c858 : 00 40 60 2c 38 28 66 00   74
c860 : 00 30 08 08 08 10 3c 00   6d
c868 : 00 00 42 77 5d 49 41 00   0d
c870 : 00 00 58 64 42 44 44 00   6a
c878 : 00 00 3d 4e 42 42 3c 00   b9
c880 : 00 00 7c 22 44 78 40 40   6d
c888 : 00 00 3c 44 22 1e 02 02   3f
c890 : 00 00 5c 62 40 40 40 00   fb
c898 : 00 01 3e 40 3c 04 78 00   76
c8a0 : 00 11 7e 08 10 10 0e 00   83
c8a8 : 00 01 42 42 42 42 3e 00   31
c8b0 : 00 00 42 42 42 24 18 00   2f
c8b8 : 00 00 41 69 6b 3e 32 00   a7
c8c0 : 00 00 46 2c 18 34 46 00   14
c8c8 : 00 00 44 44 22 1e 0c 78   96
c8d0 : 00 00 7e 04 18 21 7e 00   75
c8d8 : 1c 20 20 20 20 20 1c 00   84
c8e0 : 10 10 10 10 7c 10 38 00   28
c8e8 : 38 04 04 04 04 04 38 00   e5
c8f0 : 00 18 24 66 18 18 18 00   75
c8f8 : 00 10 30 4f 4f 30 10 00   ad
c900 : 00 00 00 00 00 00 00 00   01
c908 : 3c 3c 18 18 00 18 18 00   8d
c910 : ee 66 77 00 00 00 00 00   0f
c918 : 42 42 ff 24 ff 42 42 00   1b
c920 : 08 3e 50 3c 0a 7c 10 00   a8
c928 : 22 64 08 10 20 46 44 00   c6
c930 : 3c 42 24 28 47 42 3f 00   1f
c938 : 0e 0e 1c 00 00 00 00 00   55
c940 : 0c 08 10 20 10 08 0c 00   ca
c948 : 30 10 08 04 08 10 30 00   c5
c950 : 00 66 18 e7 18 66 00 00   3b
c958 : 00 18 18 66 18 18 00 00   7a
c960 : 00 00 00 00 1c 1c 1c 38   e4
c968 : 00 00 02 3c 40 00 00 00   75
c970 : 00 00 00 00 08 14 08 00   b2
c978 : 00 03 04 08 10 20 40 00   ff
c980 : 3d 4e 4a 52 62 62 3c 00   eb
c988 : 18 38 78 18 18 1a 3c 40   a1
c990 : 38 44 02 0c 30 40 7e 00   eb
c998 : 3c 42 04 18 04 42 3c 00   3d
c9a0 : 06 0a 12 22 7f 06 06 00   b4
c9a8 : 7e 40 7c 02 04 44 38 00   e9
c9b0 : 3c 42 40 7c 42 44 38 00   d4
c9b8 : 3e 42 04 08 08 10 10 00   5b
c9c0 : 1c 22 44 38 44 22 1c 00   cb
c9c8 : 3c 42 22 1e 04 62 3c 00   b6
c9d0 : 00 08 18 10 00 18 10 00   de
c9d8 : 00 00 1c 00 1c 1c 1c 38   63
c9e0 : 0e 18 30 50 30 18 0e 00   0c
c9e8 : 00 42 3c 00 3c 42 00 00   ee
c9f0 : 70 18 0c 0a 0c 18 70 00   f4
c9f8 : 3c 42 02 04 18 00 18 00   38
ca00 : 00 00 00 ff ff 00 00 00   00
ca08 : 18 3c 42 7e 24 42 42 00   fc
ca10 : 7c 22 24 58 24 22 3c 00   f6
ca18 : 3c 42 20 20 20 42 3c 00   87
ca20 : 78 44 22 22 22 44 78 00   ad
ca28 : 7e 40 20 38 40 20 1e 00   53
ca30 : 1e 20 40 78 40 40 40 00   85
ca38 : 3c 42 40 6c 22 24 18 00   d7
ca40 : 66 22 22 3e 22 22 66 00   d5
ca48 : 3c 00 18 18 18 18 3c 00   c1
ca50 : 1e 00 0c 04 08 48 30 00   76
ca58 : 66 24 28 30 28 28 66 00   3e
ca60 : 40 40 20 20 20 40 7e 00   ca
ca68 : 63 77 5d 49 21 22 41 00   30
ca70 : 62 72 4a 4a 46 46 42 00   87
ca78 : 3d 4e 42 42 42 42 3c 00   dc
ca80 : 7c 02 62 3c 40 20 20 00   a3
ca88 : 1c 22 24 24 44 38 0e 00   81
ca90 : 1c 22 44 78 50 4c 46 00   5e
ca98 : 3c 42 20 18 04 42 3c 00   44
caa0 : 7e 10 08 08 10 10 10 00   eb
caa8 : 44 44 22 22 22 44 38 00   00
cab0 : 44 44 42 22 22 14 08 00   ce
cab8 : 41 41 41 29 2d 37 63 00   29
cac0 : 42 66 3c 18 3c 66 42 00   47
cac8 : 42 62 22 14 18 18 18 00   e9
cad0 : 7e 02 0c 10 20 22 7c 00   5a
cad8 : 18 18 18 ff ff 18 18 18   54
cae0 : c0 c0 30 30 c0 c0 30 30   46
cae8 : 18 18 18 18 18 18 18 18   e8
caf0 : 33 33 cc cc 33 33 cc cc   23
caf8 : 33 99 cc 66 33 99 cc 66   f8
cb00 : 00 00 00 00 00 00 00 00   01
cb08 : f0 f0 f0 f0 f0 f0 f0 f0   07
cb10 : 00 00 00 00 ff ff ff ff   10
cb18 : ff 00 00 00 00 00 00 00   18
cb20 : 00 00 00 00 00 00 00 ff   20
cb28 : c0 c0 c0 c0 c0 c0 c0 c0   27
cb30 : cc cc 33 33 cc cc 33 33   fc
cb38 : 03 03 03 03 03 03 03 03   38
cb40 : 00 00 00 00 cc cc 33 33   a7
cb48 : cc 99 33 66 cc 99 33 66   ae
cb50 : 03 03 03 03 03 03 03 03   50
cb58 : 18 18 18 1f 1f 18 18 18   aa
cb60 : 00 00 00 00 0f 0f 0f 0f   24
cb68 : 18 18 18 1f 1f 00 00 00   68
cb70 : 00 00 00 f8 f8 18 18 18   70
cb78 : 00 00 00 00 00 00 ff ff   78
cb80 : 00 00 00 1f 1f 18 18 18   a8
cb88 : 18 18 18 ff ff 00 00 00   b2
cb90 : 00 00 00 ff ff 18 18 18   e2
cb98 : 18 18 18 f8 f8 18 18 18   c2
cba0 : c0 c0 c0 c0 c0 c0 c0 c0   9f
cba8 : e0 e0 e0 e0 e0 e0 e0 e0   a7
cbb0 : 07 07 07 07 07 07 07 07   b0
cbb8 : ff ff 00 00 00 00 00 00   b7
cbc0 : ff ff ff 00 00 00 00 00   bf
cbc8 : 00 00 00 00 00 ff ff ff   c8
cbd0 : 01 03 06 6c 78 70 60 00   ef
cbd8 : 00 00 00 00 f0 f0 f0 f0   15
cbe0 : 0f 0f 0f 0f 00 00 00 00   1d
cbe8 : 18 18 18 f8 f8 00 00 00   c1
cbf0 : f0 f0 f0 f0 00 00 00 00   b3
cbf8 : f0 f0 f0 f0 0f 0f 0f 0f   7e
cc00 : c3 bd 66 5e 66 bd c3 ff   6b
cc08 : ff ff c3 fd c1 bd c1 ff   c8
cc10 : ff bf bf 83 bd bd 83 ff   27
cc18 : ff ff c3 bf bf bf c3 ff   09
cc20 : ff fd fd c1 bd bd c1 ff   a6
cc28 : ff ff c3 bd 81 bf c3 ff   f4
cc30 : ff f1 ef c3 ef ef ef ff   da
cc38 : ff ff c1 bd bd c1 fd 83   48
cc40 : ff bf bf 83 bd bd bd ff   40
cc48 : ff e7 ff e7 f7 f7 e3 ff   06
cc50 : ff fd ff fd fd fd fd c3   5d
cc58 : ff bf bf bb 87 b7 b9 ff   bb
cc60 : ff e7 f7 f7 f7 f7 e3 ff   1e
cc68 : ff ff dd 88 a2 b6 be ff   ca
cc70 : ff ff 83 bd bd bd bd ff   c8
cc78 : ff ff c3 bd bd bd c3 ff   f8
cc80 : ff ff 83 bd bd 83 bf bf   8e
cc88 : ff ff c1 bd bd c1 fd fd   8d
cc90 : ff ff 83 bd bf bf bf ff   20
cc98 : ff ff c1 bf c3 fd 83 ff   39
cca0 : ff ef 83 ef ef ef f1 ff   bc
cca8 : ff ff bd bd bd bd c1 ff   9f
ccb0 : ff ff bd bd bd db e7 ff   30
ccb8 : ff ff be b6 a2 c1 c9 ff   9d
ccc0 : ff ff bd db e7 db bd ff   fe
ccc8 : ff ff bd bd bd c1 fb 87   d7
ccd0 : ff ff 83 f7 ef df 81 ff   b3
ccd8 : c3 cf cf cf cf cf c3 ff   fb
cce0 : f3 ed cf 83 cf 9d 03 ff   24
cce8 : c3 f3 f3 f3 f3 f3 c3 ff   0e
ccf0 : ff e7 c3 81 e7 e7 e7 e7   31
ccf8 : ff ef cf 80 80 cf ef ff   39
cd00 : ff ff ff ff ff ff ff ff   ff
cd08 : e7 e7 e7 e7 ff ff f7 ff   b9
cd10 : 99 99 99 ff ff ff ff ff   dc
cd18 : 99 99 00 99 00 99 99 ff   e4
cd20 : e7 c1 9f c3 f9 83 e7 ff   a3
cd28 : 9d 99 f3 e7 cf 99 b9 ff   3c
cd30 : c3 99 c3 c7 98 99 c0 ff   03
cd38 : f9 f3 e7 ff ff ff ff ff   25
cd40 : f3 e7 cf cf cf e7 f3 ff   21
cd48 : cf e7 f3 f3 f3 e7 cf ff   44
cd50 : ff 99 c3 00 c3 99 ff ff   16
cd58 : ff e7 e7 81 e7 e7 ff ff   33
cd60 : ff ff ff ff ff e7 e7 cf   dd
```

## Listing 1. (Fortsetzung)

```
cd68 : ff ff ff 81 ff ff ff ff   97
cd70 : ff ff ff ff ff e7 e7 ff   4d
cd78 : ff fc f9 f3 e7 cf 9f ff   6e
cd80 : c3 99 91 89 99 99 c3 ff   1b
cd88 : e7 e7 c7 e7 e7 e7 81 ff   15
cd90 : c3 99 f9 f3 cf 9f 81 ff   1d
cd98 : c3 99 f9 e3 f9 99 c3 ff   9e
cda0 : f9 f1 e1 99 80 f9 f9 ff   fd
cda8 : 81 9f 83 f9 f9 99 c3 ff   94
cdb0 : c3 99 9f 83 99 99 c3 ff   0e
cdb8 : 81 99 f3 e7 e7 e7 e7 ff   5d
cdc0 : c3 99 99 c3 99 99 c3 ff   a4
cdc8 : c3 99 99 c1 f9 99 c3 ff   72
cdd0 : ff ff e7 ff ff e7 ff ff   08
cdd8 : ff ff e7 ff ff e7 e7 cf   4f
cde0 : f1 e7 cf 9f cf e7 f1 ff   b1
cde8 : ff ff 81 ff 81 ff ff ff   5f
cdf0 : 8f e7 f3 f9 f3 e7 8f ff   6c
cdf8 : c3 99 f9 f3 e7 ff e7 ff   a3
ce00 : ff ff ff 00 00 ff ff ff   ff
ce08 : c3 bd bd 81 bd bd bd ff   0a
ce10 : 83 bd bd 83 bd bd 83 ff   29
ce18 : c3 bd bf bf bf bd c3 ff   9b
ce20 : 87 bb bd bd bd bb 87 ff   84
ce28 : c1 bf bf 87 bf bf c1 ff   ab
ce30 : c1 bf bf 87 bf bf bf ff   ab
ce38 : c3 bd bf b1 bd bd c3 ff   d9
ce40 : bd bd bd 81 bd bd bd ff   3c
ce48 : e3 f7 f7 f7 f7 f7 e3 ff   f2
ce50 : f1 fb fb fb fb bb c7 ff   7a
ce58 : bb b7 af 9f af b7 bb ff   76
ce60 : bf bf bf bf bf bf c1 ff   e8
ce68 : be 9c aa b6 be be be ff   d3
ce70 : bd 9d 8d a5 b1 b9 bd ff   f4
ce78 : c3 bd bd bd bd bd c3 ff   1a
ce80 : 83 bd bd 83 bf bf bf ff   bb
ce88 : c3 bd bd bd bd c3 f1 ff   13
ce90 : 83 bd bd bb 87 bb bd ff   26
ce98 : c3 bd bf c3 fd bd c3 ff   7f
cea0 : c1 f7 f7 f7 f7 f7 f7 ff   79
cea8 : bd bd bd bd bd bd c3 ff   44
ceb0 : bd bd bd bd bd db e7 ff   cd
ceb8 : be be be b6 a2 88 9c ff   3d
cec0 : bd db e7 e7 e7 db bd ff   b6
cec8 : bd bd bd c3 f7 f7 ef ff   4b
ced0 : 81 fd fb e7 df bf 81 ff   4e
ced8 : e7 e7 e7 00 00 e7 e7 e7   5b
cee0 : 3f 3f cf cf 3f 3f cf cf   79
cee8 : e7 e7 e7 e7 e7 e7 e7 e7   e7
cef0 : cc cc 33 33 cc cc 33 33   bc
cef8 : cc 66 33 99 cc 66 33 99   f7
cf00 : ff ff ff ff ff ff ff ff   ff
cf08 : 0f 0f 0f 0f 0f 0f 0f 0f   08
cf10 : ff ff ff ff 00 00 00 00   0f
cf18 : 00 ff ff ff ff ff ff ff   18
cf20 : ff ff ff ff ff ff ff 00   1f
cf28 : 3f 3f 3f 3f 3f 3f 3f 3f   28
cf30 : 33 33 cc cc 33 33 cc cc   63
cf38 : fc fc fc fc fc fc fc fc   37
cf40 : ff ff ff ff 33 33 cc cc   d8
cf48 : 33 66 cc 99 33 66 cc 99   e1
cf50 : fc fc fc fc fc fc fc fc   4f
cf58 : e7 e7 e7 e0 e0 e7 e7 e7   05
cf60 : ff ff ff ff f0 f0 f0 f0   9b
cf68 : e7 e7 e7 e0 e0 ff ff ff   67
cf70 : ff ff ff 07 07 e7 e7 e7   6f
cf78 : ff ff ff ff ff ff 00 00   77
cf80 : ff ff ff e0 e0 e7 e7 e7   57
cf88 : e7 e7 e7 00 00 ff ff ff   5d
cf90 : ff ff ff 00 00 e7 e7 e7   3d
cf98 : e7 e7 e7 07 07 e7 e7 e7   6d
cfa0 : 3f 3f 3f 3f 3f 3f 3f 3f   a0
cfa8 : 1f 1f 1f 1f 1f 1f 1f 1f   a8
cfb0 : f8 f8 f8 f8 f8 f8 f8 f8   af
cfb8 : 00 00 ff ff ff ff ff ff   b8
cfc0 : 00 00 00 ff ff ff ff ff   c0
cfc8 : ff ff ff ff ff 00 00 00   c7
cfd0 : fe fc f9 93 87 8f 9f ff   b1
cfd8 : ff ff ff ff 0f 0f 0f 0f   9a
cfe0 : f0 f0 f0 f0 ff ff ff ff   a2
cfe8 : e7 e7 e7 07 07 ff ff ff   0e
cff0 : 0f 0f 0f 0f ff ff ff ff   2d
cff8 : 0f 0f 0f 0f f0 f0 f0 f0   71
```

## Listing 2. »B«

```
programm : b                  0334 0402
---------------------------------------

0334 : ad 02 dd 09 03 8d 02 dd   db
033c : ad 00 dd 29 fc 09 00 8d   b9
0344 : 00 dd ad 18 d0 29 0f 09   46
034c : 00 8d 18 d0 a9 c0 8d 88   1b
0354 : 02 ad 18 d0 29 f0 09 02   8f
035c : 8d 18 d0 20 44 e5 60 ad   7e
0364 : 02 dd 09 03 8d 02 dd ad   b3
036c : 00 dd 29 fc 09 03 8d 00   24
0374 : dd ad 18 d0 29 0f 09 10   97
037c : 8d 18 d0 a9 04 8d 88 02   51
0384 : ad 18 d0 29 f0 09 04 8d   19
038c : 18 d0 20 44 e5 60 00 00   fe
0394 : 00 00 00 00 00 00 00 00   95
039c : 00 00 00 00 00 00 00 00   9d
03a4 : 00 00 00 00 00 00 00 00   a5
03ac : 00 00 00 00 00 00 00 00   ad
03b4 : 00 00 00 00 00 00 00 00   b5
03bc : 00 00 00 00 00 00 00 00   bd
03c4 : 00 00 00 00 00 00 00 00   c5
03cc : 00 00 00 00 00 00 00 00   cd
03d4 : 00 00 00 00 00 00 00 00   d5
03dc : 00 00 00 00 00 00 00 00   dd
03e4 : 00 00 00 00 00 00 00 00   e5
03ec : 00 00 00 00 00 00 00 00   ed
03f4 : 00 00 00 00 00 00 00 00   f5
03fc : 00 00 00 00 20 10 12 0f   e5
```

## Listing 3. »The Sword«

```
10 POKE 808,234                               <242>
14 PRINT"{CLR,DOWN,SPACE}LOADING..."          <247>
15 SYS 57812"A",8,1:POKE 780,0:SYS 65493      <134>
16 SYS 57812"B",8,1:POKE 780,0:SYS 65493      <137>
20 POKE 53280,6:POKE 53281,6:PRINT CHR$(14
   7)CHR$(8)CHR$(14):POKE 650,0:POKE 53269
   ,0                                         <075>
21 SYS 820                                    <121>
50 PRINT"{CLR,4DOWN,LIG.RED,7SPACE}PINK AR
   TS PROUDLY PRESENTS:"                      <120>
60 PRINT"{2DOWN,13SPACE,WHITE}>{BLACK,SPAC
   E}THE SWORD{SPACE,WHITE}<{BLACK}"          <209>
65 PRINT"{DOWN,12SPACE}@ 1986 A.SOMMER"       <101>
67 PRINT"{DOWN,15SPACE}AND S.ADOMAT"          <057>
70 FOR T=1 TO 1000:NEXT                       <073>
75 PRINT"{3DOWN,15SPACE}THANKS TO"            <163>
80 PRINT"{DOWN,7SPACE}BIRTHE,CLAUDIA,THE G
   IRLS..."                                   <061>
90 FOR T=1 TO 1000:NEXT                       <093>
95 PRINT"{2DOWN,15SPACE}(WAITING)"            <083>
100 AV=15:DIM VE$(AV)                         <017>
101 AO=32:DIM NO$(AO),O$(AO),O(AO)            <087>
102 AR=41:DIM RA$(AR),DU(AR,6),KR$(AR)        <048>
103 AU=18:DIM AU$(AU)                         <163>
104 AF=9:DIM FL(AF)                           <211>
105 SP=1:F1=0:F2=15:F3=15:F0=6:ZU=0:PO=0      <178>
106 A2=28:DIM V2$(A2)                         <129>
107 AZ=4:DIM ZA$(AZ)                          <026>
131 DATA NIMM,LEGE,GEHE,BETRETE,VERLASSE,T
    OETE,RUDERE,LEG                           <071>
132 DATA VERLIERE,LESE,LIES,HEIRATE,GEH,BE
    STEIGE,UNTERSUCHE                         <117>
141 DATA"/","NORDEN",0                        <064>
142 DATA"/","SUEDEN",0                        <030>
143 DATA"/","WESTEN",0                        <187>
144 DATA"/","OSTEN",0                         <188>
145 DATA"/","OBEN",0                          <015>
146 DATA"/","UNTEN",0                         <186>
147 DATA"/","MIR",0                           <005>
148 DATA"/","MICH",0                          <230>
149 DATA"DEN HEILIGEN MANN","MANN",2          <084>
150 DATA"EIN BOOT","BOOT",3                   <197>
151 DATA"KEIN BOOT","BOOT",9                  <082>
152 DATA"DIE LIEBLICHE BIRTHE","BIRTHE",10    <237>
153 DATA"DIE NIEDLICHE CLAUDIA","CLAUDIA",
    10                                        <126>
154 DATA"DIE HEISSBLUETIGE CHRISTIANE","CH
    RISTIANE",10                              <004>
155 DATA"DIE RESERVIERTE MEIKE","MEIKE",10    <085>
156 DATA"DIE NETTE SANDRA","SANDRA",10        <212>
157 DATA"VIELE GROSSE UND KLEINE STEINE","
    STEINE",11                                <096>
158 DATA"BAEUME","BAUEME",14                  <042>
159 DATA"BAEUME","BAEUME",15                  <115>
160 DATA"EINE KLEINE,ALTE HUETTE","HUETTE"
    ,16                                       <008>
161 DATA"EINEN MORGENSTERN","MORGENSTERN",
    19                                        <064>
162 DATA"EINEN KOSTBAREN SCHILD","SCHILD",
    19                                        <005>
163 DATA"EIN VERGILBTES BLATT PAPIER","BLA
    TT",24                                    <173>
164 DATA"EINEN (DEN!) ZWERG","ZWERG",32       <189>
165 DATA"DEN DRACHEN,DEN ICH TOETEN MUSS",
    "DRACHEN",33                              <043>
166 DATA"GAR NICHTS,NUR LAUTER UEBELRICHEN
    DEN{4SPACE}NEBEL","NEBEL",35              <248>
167 DATA"KEINE RETTUNG FUER MICH","RETTUNG
    ",41                                      <072>
168 DATA"/","PAPIER",0                        <226>
169 DATA"/","FRAU",0                          <102>
170 DATA"/","ALLES",0                         <235>
171 DATA"DAS ZAUBERSCHWERT","ZAUBERSCHWERT
    ",33                                      <185>
172 DATA"EINE LAMPE","LAMPE",-1               <037>
201 DATA"VOR DEM HEILIGEN TURM.","VORM HEI
    LIGEN TURM",2,0,0,0,0,0                   <079>
202 DATA"IM HEILIGEN TURM.KALT HIER...","I
    M TURM",9,1,5,6,10,3                      <154>
203 DATA"IM KELLER.HIER FLIESST EIN{6SPACE
    }FLUSS.","KELLER",0,0,0,0,2,0             <036>
204 DATA"IM BOOT.","IM BOOT",0,0,0,0,0,0      <232>
205 DATA"IM MOOR.","MOOR",8,0,0,2,0,0         <106>
206 DATA"AM MEER.","AM MEER",0,0,2,0,0,0      <223>
207 DATA"IM BOOT.","IM BOOT",0,0,0,0,0,0      <235>
208 DATA"IM MOOR.","MOOR",0,0,0,0,0,0         <229>
209 DATA"AM FLUSS.","AM FLUSS",11,2,0,0,0,
    0                                         <186>
210 DATA"IM HAREM DES HEILIGEN MANNES.","I
    M HAREM",0,0,0,0,0,2                      <038>
211 DATA"IM GEBIRGE.","GEBIRGE",11,11,11,1
    1,12,9                                    <113>
212 DATA"AUF DEM GIPFEL DER GENUESSE.","GI
    PFEL",0,0,0,0,0,0                         <117>
```

**Listing 3. »The Sword« (Fortsetzung)**

```
213 DATA"AUF EINER SANDBANK.(MIST!)","SAND
    BANK",0,0,0,0,0,0                        <229>
214 DATA"IM WALD.","WALD",15,0,14,14,0,0   <159>
215 DATA"IM WALD.","WALD",16,14,15,14,0,0  <133>
216 DATA"IM WALD VOR EINER HUETTE.","VOR E
    INER HUETTE",15,15,15,14,0,0              <109>
217 DATA"IN DER KLEINEN HUETTE.","HUETTE",
    0,0,0,0,0,0                               <149>
218 DATA"AM SEE KALTOR.","KALTOR",21,0,41,
    19,0,0                                    <235>
219 DATA"AUF EINER EBENE.","EBENE",0,0,18,
    0,0,0                                     <081>
220 DATA"AUF EINER WEITEN EBENE.","EBENE",
    0,0,0,21,0,0                              <163>
221 DATA"AN EINEM FLUSS.","FLUSS",22,18,20
    ,0,0,0                                    <149>
222 DATA"AM FLUSS.","FLUSS",23,21,20,0,0,0 <067>
223 DATA"AM FLUSS.","FLUSS",24,22,20,0,0,0 <201>
224 DATA"AM RAND DES GEBIRGES.","GEBIRGSRA
    ND",26,23,23,23,0,0                       <082>
225 DATA"IM GEBIRGE.","GEBIRGE",28,0,39,26
    ,0,0                                      <107>
226 DATA"IM GEBIRGE.","GEBIRGE",29,24,25,2
    7,0,0                                     <101>
227 DATA"IN EINEM TAL.","TAL",30,0,26,0,0,
    0                                         <043>
228 DATA"IM GEBIRGE.","GEBIRGE",0,25,39,29
    ,0,0                                      <031>
229 DATA"IN EINEM TAL.","TAL",31,26,28,30,
    0,0                                       <117>
230 DATA"IM GEBIRGE.","GEBIRGE",0,27,30,0,
    0,0                                       <145>
231 DATA"IN EINEM TAL.","TAL",0,29,0,0,0,3
    2                                         <160>
232 DATA"IN EINER DUESTEREN HOEHLE.","HOEH
    LE",0,0,0,0,31,0                          <137>
233 DATA"TIEF IN DER HOEHLE DES LOEWEN...A
    EHH DRACHEN,NATUERLICH!"                  <006>
234 DATA"HOEHLE",0,0,0,0,34,0               <225>
235 DATA"IN DER HOEHLE DES DRACHEN.","HOEH
    LE",0,0,0,35,0,33                         <064>
236 DATA"AM FUSSE DES VETUS.","VETUS",0,40
    ,34,36,0,0                                <172>
237 DATA"AUF EINER EBENE.FERN IM WESTEN{2S
    PACE}RAUCHT DER VETUS.","EBENE"           <033>
238 DATA 0,0,35,37,0,0                       <230>
239 DATA"AUF EINER EBENE.","EBENE",39,38,3
    6,38,0,0                                  <157>
240 DATA"AUF EINER EBENE.","EBENE",39,37,3
    7,39,0,0                                  <229>
241 DATA"AUF EINER EBENE.","EBENE",38,25,3
    8,28,0,0                                  <236>
242 DATA"AUF EINER EBENE.","EBENE",35,40,4
    0,40,0,0                                  <138>
243 DATA"UNTER WASSER...","UNTER WASSER",0
    ,0,0,0,0,0                                <119>
300 DATA DER,DIE,DAS,DEN, ,UEBER,UNTER,AUF
    ,IN,MIT,DEM,DURCH,UND,NACH                <214>
301 DATA BITTE,DOCH,HEILIGEN,ZUR             <122>
501 DATA 0,0,0,0,0,0,0,0,0                   <214>
601 DATA INVENTORY,HILFE,ENDE,SIEH,LOOK,DA
    NKE,SAVE,LOAD,STORE,RESTORE,FARBE         <064>
602 DATA NORDEN,SUEDEN,WESTEN,OSTEN,OBEN,U
    NTEN,SCORE,SPIELSTAND,SAGE,SCHLAFE        <019>
603 DATA SPEICHERE,LADE,ANLEITUNG,SPIELANL
    EITUNG,NORMAL,NEU,WARTE                   <128>
650 REM ZAUBERSPRUECHE                       <143>
651 DATA EMMIWHCS,KCEURUZ,"EHCARD ETEOT","
    THCIL"                                    <075>
700 DATA 133,95,134,96,76,191,163            <138>
800 FOR T=1 TO AV:READ VE$(T):NEXT           <009>
801 FOR T=1 TO AO:READ NO$(T),O$(T),O(T)     <108>
802 NEXT                                     <050>
803 FOR T=1 TO AR:READ RA$(T),KR$(T)         <093>
804 FOR X=1 TO 6:READ DU(T,X):NEXT X,T       <131>
805 FOR T=1 TO AU:READ AU$(T):NEXT           <146>
806 FOR T=1 TO AF:READ FL(T):NEXT           <169>
807 FOR T=1 TO A2:READ V2$(T):NEXT          <231>
808 FOR T=1 TO AZ:READ ZA$(T):ZA$(T)=CHR$(
    34)+ZA$(T)+CHR$(34):NEXT                  <034>
810 FOR T=704 TO 710:READ A:POKE T,A:NEXT    <082>
820 POKE 88,0:POKE 89,192:POKE 90,0:POKE 9
    1,192:POKE 780,0:POKE 781,160:SYS 704     <023>
825 POKE 88,0:POKE 89,0:POKE 90,0:POKE 91,
    0:POKE 780,0:POKE 781,224:SYS 704         <207>
830 POKE 1,53                                <095>
900 PRINT"{UP,7SPACE}NORMALER ZEICHENSATZ?
    {2SPACE}@{LEFT}";                         <073>
920 GET A$:IF A$<>"N"AND A$<>"J"THEN 920     <175>
930 PRINT A$                                 <090>
931 FOR T=1 TO 500:NEXT                      <241>
940 IF A$="J"THEN GOSUB 56000:PRINT CHR$(1
    4):F0=0:F1=5:F2=3:F3=5                    <219>
945 POKE 53280,F0:POKE 53281,F0              <041>
1000 POKE 59639,0                            <066>
1005 PRINT"{CLR}"                            <231>
1006 DATA"NORDEN","SUEDEN","WESTEN","OSTEN
     ","OBEN","UNTEN"                         <192>
1010 FOR T=1 TO 6:READ RI$(T):NEXT           <064>
1020 PRINT CHR$(147)                         <031>
1080 GOSUB 53000:PRINT:POKE 646,F1:IF SP=2
      THEN FL(2)=-1                           <168>
1090 PRINT"ICH BIN ";RA$(SP)                 <063>
1100 PRINT"I CAN SEE:"                       <043>
1110 FL(1)=-1:FOR T=1 TO AO                  <061>
1120 IF O(T)=SP THEN PRINT NO$(T)".":FL(1)
     =0                                       <139>
1130 NEXT                                    <124>
1140 IF FL(1)THEN PRINT"NICHTS."             <200>
1150 PRINT"HIER GEHT'S NACH:"                <155>
1160 G=0:FOR T=1 TO 6                        <109>
1170 IF DU(SP,T)<>0 THEN PRINT RI$(T)",";:
     G=G+1                                    <247>
1180 NEXT                                    <174>
1190 IF G<1 THEN PRINT"NIRGENDWOHIN ";       <200>
1200 PRINT"{LEFT}."                          <085>
1230 IF FL(2)AND O(9)=2 THEN O(9)=0:GOSUB
     57000:GOTO 1080                          <245>
1300 IF SP=8 THEN PRINT"{DOWN}HILFE,ICH VE
     RSINKE IM MOOR...{DOWN}":GOTO 54000      <213>
1310 IF SP=13 THEN PRINT"{DOWN}TJA,DAS ABE
     NTEUER IST LEIDER ZUENDE."               <134>
1320 IF SP=13 THEN PRINT"ICH VERHUNGERE LA
     NGSAM,UND DENKE"                         <055>
1330 IF SP=13 THEN PRINT"AN DIE ZEIT,DIE I
     CH MIT BIRTHE HAETTE"                    <023>
1340 IF SP=13 THEN PRINT"VERBRINGEN KOENNE
     N.":PRINT:GOTO 54000                     <111>
1370 :                                       <076>
1500 ZU=ZU+1:IF FL(5)AND NOT FL(8)THEN PO=
     PO+1:FL(8)=-1                            <126>
1501 IF SP<>41 THEN FL(7)=0                  <230>
1502 IF SP=41 THEN FL(7)=FL(7)+1             <077>
1503 IF FL(7)>2 THEN PRINT"ICH ERTRINKE...
     BLUBB...":PRINT:GOTO 54000               <177>
1505 GOSUB 50000:POKE 646,F2                 <107>
1510 IF LEN(X$)>2 THEN 1600                  <023>
1520 IF X$="N"AND DU(SP,1)<>0 THEN SP=DU(S
     P,1):PRINT"BITTE.":GOTO 1080             <190>
1521 IF X$="S"AND DU(SP,2)<>0 THEN SP=DU(S
     P,2):PRINT"NA GUT.":GOTO 1080            <237>
1523 IF X$="W"AND DU(SP,3)<>0 THEN SP=DU(S
     P,3):PRINT"GOOOOD.":GOTO 1080            <243>
1524 IF X$="O"AND DU(SP,4)<>0 THEN SP=DU(S
     P,4):PRINT"NA KLAR.":GOTO 1080           <205>
1525 IF X$="OB"AND DU(SP,5)<>0 THEN SP=DU(
     SP,5):PRINT"ABER BITTE.":GOTO 1080       <139>
1526 IF X$="U"AND DU(SP,6)<>0 THEN SP=DU(S
     P,6):PRINT"BITTESCHOEN.":GOTO 1080       <154>
1527 IF X$="I"THEN 31000                     <144>
1599 PRINT"WRONG INPUT!":GOTO 1500           <019>
1600 ER=0:GOSUB 51000                        <083>
1610 IF ER=1 THEN 1500                       <036>
1615 ER=0:GOSUB 52000                        <100>
1616 IF ER=1 THEN 1500                       <042>
1620 IF V2<>0 THEN 30000                     <180>
1630 IF O1=28 THEN O1=23                     <074>
1640 IF O1=18 AND SP=15 THEN O1=19           <132>
1650 IF O1=10 AND SP=9 THEN O1=11            <232>
1700 ON V GOTO 2500,2000,3000,3500,4000,45
     00,5000,2000,6500,7000,7000,7500         <113>
1710 ON V-12 GOTO 3000,6000,8000             <084>
1800 GOTO 1500                               <042>
2000 IF O2<>0 THEN PRINT"IMMER DER REIHE N
     ACH!":GOTO 1500                          <203>
2005 IF O1=30 THEN 2100                      <043>
2010 IF NOT O(O1)THEN PRINT"HABE ICH NICHT
      DABEI!":GOTO 1500                       <199>
2020 O(O1)=SP:PRINT"GELEGT!":GOTO 1500       <074>
2100 GL=0:FOR T=1 TO AO                      <225>
2110 IF O(T)=-1 THEN O(T)=SP:PRINT O$(T)":
     GELEGT":GL=1                             <076>
```

```
2120 NEXT                                         <098>
2130 IF GL=1 THEN 1500                            <223>
2140 PRINT"HATTE LEIDER NOTHING DABEI!":GO
     TO 1500                                      <074>
2500 IF O1=30 THEN 2800                           <058>
2501 IF O2=29 THEN O2=0:GOTO 7500                 <094>
2502 IF O1=8 THEN PRINT"DU HASST DICH DOCH
     SCHON!":GOTO 1500                            <062>
2503 IF O1=7 THEN PRINT"LERN ERSTMAL DEUTS
     CH!":GOTO 1500                               <138>
2504 IF O(O1)=-1 THEN PRINT"HAB ICH DOCH S
     CHON!":GOTO 1500                             <046>
2505 IF O(O1)<>SP THEN PRINT"SEHE ICH HIER
      NICHT!":GOTO 1500                           <240>
2506 IF O1=11 THEN PRINT"DA OBEN STEHT KEI
     N BOOT!":GOTO 1500                           <094>
2507 IF O1<12 THEN PRINT"DAS GEHT NUN WIRK
     LICH NICHT!":GOTO 1500                       <057>
2508 IF O1<17 THEN PRINT"DU DENKST AUCH NU
     R AN DAS EINE!":GOTO 1500                    <034>
2509 IF O1=17 THEN PRINT"SO VIELE...KANN M
     ICH NICHT ENTSCHEIDEN!":GOTO 1500            <205>
2510 IF O1=18 OR O1=19 THEN PRINT"BIN DOCH
      KEIN RIESE,NUR EIN KLEINER C64!";:GO
     TO 1500                                      <143>
2511 IF O1=20 THEN PRINT"DAS SOLLTEST DU D
     IR NOCHMAL UEBERLEGEN!":GOTO 1500            <081>
2512 IF O1>23 AND O1<31 THEN PRINT"NEIN,DA
     S UEBERSTEIGT MEINE KOMPETENZEN!":GOT
     O 1500                                       <043>
2530 IF O1=31 AND NOT FL(6)THEN PRINT"DER
     DRACHE VERFEUERT DICH IM VULKAN!":GOT
     O 54000                                      <064>
2600 O(O1)=-1:PRINT"ALLRIGHT,GETAKET!"            <105>
2610 IF O(31)THEN FL(5)=-1                        <133>
2630 GOTO 1500                                    <110>
2800 GN=0:FOR T=1 TO AO                           <177>
2810 IF O(T)=SP THEN GOSUB 2850                   <241>
2820 NEXT                                         <036>
2830 IF GN=0 THEN PRINT"NICHTS DA!"               <093>
2840 GOTO 1500                                    <066>
2850 IF T<>21 AND T<>22 AND T<>23 AND T<>3
     2 THEN 2900                                  <247>
2860 PRINT O$(T)":GENOMMEN":O(T)=-1:GN=1:R
     ETURN                                        <007>
2900 IF T=31 AND FL(6)=-1 THEN FL(5)=-1:GO
     TO 2860                                      <214>
2905 IF T=31 AND NOT FL(6)THEN O1=31:GOTO
     2530                                         <041>
2910 RETURN                                       <174>
3000 IF O2<>0 THEN PRINT"EINS AUF EINMAL!"
     :GOTO 1500                                   <173>
3010 IF O1=1 THEN X$="N":GOTO 1520                <225>
3020 IF O1=2 THEN X$="S":GOTO 1521                <132>
3030 IF O1=3 THEN X$="W":GOTO 1523                <030>
3040 IF O1=4 THEN X$="O":GOTO 1524                <240>
3050 IF O1=5 THEN X$="OB":GOTO 1525               <048>
3060 IF O1=6 THEN X$="U":GOTO 1526                <245>
3070 PRINT"NIT MOEGLICH...":GOTO 1500             <014>
3500 IF O2<>0 THEN PRINT"NO,MAN!":GOTO 150
     0                                            <011>
3510 IF O1=10 THEN 6000                           <156>
3520 IF O(O1)<>SP THEN PRINT"SEH ICK NICH!
     ":GOTO 1500                                  <138>
3530 IF O1<>20 THEN PRINT"SOWAS KANN ICH N
     ICHT BETRETEN!":GOTO 1500                    <010>
3540 PRINT"BIN DRIN!":SP=17:PO=PO+1:GOTO 1
     080                                          <204>
4000 IF O2<>0 THEN PRINT"SO NICHT!":GOTO 1
     500                                          <251>
4010 IF O1<>10 AND O1<>20 THEN PRINT"DAS G
     EHT NICHT!":GOTO 1500                        <223>
4020 IF O1=20 AND SP=17 THEN PRINT"SIE LAE
     SST MICH NICHT RAUS!":GOTO 1500              <206>
4030 IF SP<>4 AND SP<>7 AND O1<>10 THEN PR
     INT"VERSTEH' DICH NICHT!":GOTO 1500          <253>
4040 IF SP=4 THEN PRINT"HILFE,ICH ERTRINKE
     ...":GOTO 54000                              <042>
4050 SP=14:PRINT"GUTE WAHL!":GOTO 1080            <167>
4100 PRINT"LEIDER UNMOEGLICH!":GOTO 1500          <149>
4500 IF O1=8 THEN PRINT"SELBSTMORD IST UNT
     ER STRAFE VERBOTEN!":GOTO 1500               <048>
4501 IF O1=7 THEN PRINT"LERN ERSTMAL RICHT
     IG DEUTSCH!":GOTO 1500                       <194>
4505 IF O(O1)<>SP OR O2<>0 AND O2<>21 THEN
      PRINT"VERSTEHE ICH NICHT.":GOTO 1500        <224>
4510 IF O1=9 OR O1=24 OR O1=25 OR O1>11 AN
     D O1<17 THEN 4600                            <012>
4530 PRINT"EIN DING KANN MAN NICHT TOETEN!
     ":GOTO 1500                                  <021>
4600 IF O2=21 THEN PRINT"DAMI GEHTS NICHT!
     ":GOTO 1500                                  <156>
4609 PRINT"WOMIT DENN?"                           <175>
4610 PRINT"UND UEBERHAUPT:MORDEN IST GAR N
     ICHT "                                       <249>
4620 PRINT"FEIN!":GOTO 1500                       <162>
5000 IF O1>4 OR O2<>0 THEN PRINT"SO EIN BL
     OEDSINN!":GOTO 1500                          <115>
5010 IF SP<>7 AND SP<>4 THEN PRINT"OHNE BO
     OT UND OHNE WASSER???":GOTO 1500             <112>
5020 IF SP=7 AND O1=2 THEN SP=4:PRINT"ALLR
     IGHT!":GOTO 1080                             <026>
5030 IF SP=4 AND O1=1 THEN SP=7:PRINT"BITT
     ESCHOEN!":GOTO 1080                          <050>
5040 IF O1=3 THEN SP=13:PRINT"SELBST SCHUL
     D!":GOTO 1080                                <112>
5050 PRINT"DAHIN KANN ICH NICHT RUDERN!":G
     OTO 1500                                     <245>
6000 IF SP<>3 THEN PRINT"DAS GEHT NICHT!":
     GOTO 1500                                    <094>
6010 IF O1<>10 THEN PRINT"WIE BITTE?":GOTO
      1500                                        <188>
6020 SP=4:PRINT"O.K.":PO=PO+1:GOTO 1080           <194>
6100 PRINT"LEIDER UNMOEGLICH!":GOTO 1500          <115>
6500 IF O1=0 THEN PRINT"WAS DENN?":GOTO 15
     00                                           <121>
6510 IF NOT O(O1)THEN PRINT"HAB ISCH NISCH
     !":GOTO 1500                                 <034>
6520 PRINT"ICH BIN DOCH NICHT SO BLOED UND
      VER-"                                       <125>
6530 PRINT"LIER WAS!"                             <227>
6540 PRINT"SCHON AUS PRINZIP NICHT!":GOTO
     1500                                         <111>
7000 IF O2<>28 AND O2<>0 THEN PRINT"WHAT,T
     O THE DONNERWETTER?":GOTO 1500               <216>
7005 IF NOT O(O1)THEN PRINT"HAB ICH NICHT!
     ":GOTO 1500                                  <035>
7010 IF O1<>23 THEN PRINT"VOELLIG UNLESERL
     ICH!":GOTO 1500                              <221>
7020 PRINT"OHH NEIN,EIN WERBE-ZETTEL..."          <054>
7030 PRINT"SOLL ICH IHN WIRKLICH LESEN?":G
     OSUB 50000:POKE 646,F2                       <145>
7040 IF LEFT$(X$,1)="N"THEN PRINT"GUT SO!"
     :GOTO 1500                                   <243>
7041 IF LEFT$(X$,1)<>"J"THEN 7030                 <095>
7050 PRINT"DORT STEHT:"                           <144>
7080 PRINT CHR$(34);:POKE 646,F1                  <049>
7090 PRINT"PINK SCHWERTPUTZER FUER IHR SCH
     WERT,"                                       <006>
7100 PRINT" JETZT IM SONDERANGEBOT BY ARTH
     URS "                                        <240>
7110 PRINT" RITTER- UND KNAPPENSHOP!"             <065>
7111 PRINT" NUR 19,50.-";                         <200>
7119 POKE 646,F2                                  <125>
7120 PRINT CHR$(34):GOTO 1500                     <246>
7500 IF O1=9 THEN PRINT"WAS BIST DU DENN F
     UER EINER?":GOTO 1500                        <089>
7510 IF O(O1)<>SP THEN PRINT"SEHE ICH HIER
      NICHT!":GOTO 1500                           <165>
7520 IF O1<12 OR O1>16 THEN PRINT"SO EIN S
     CHWACHSINN!":GOTO 1500                       <023>
7530 IF O1=15 THEN 7700                           <016>
7535 IF O1=14 THEN 7900                           <157>
7540 IF FL(5)=0 THEN 7800                         <182>
7550 PRINT"GUT,"NO$(O1)" IST NUN MEINE":PR
     INT"FRAU,";:PO=PO+1                          <068>
7555 PRINT"UND IHR LIEBT EUCH UND WERDET"         <253>
7556 PRINT"GLUECKLICH SEIN BIS AN EUER LEB
     ENSENDE!"                                    <212>
7557 PRINT"ODER?":PRINT                           <107>
7560 POKE 646,F1                                  <057>
7570 PRINT"BRAVO,DU HASST MICH GESCHLAGEN,
     DAS"                                         <073>
7580 PRINT"ADVENTURE IST GELOEST!"                <129>
7585 PRINT"DU ERHAELST"PO"VON 9 PUNKTEN!"         <176>
7590 PRINT"ABER FREU DICH NICHT ZU FRUEH,"        <207>
7595 PRINT"TEIL 2 IST SCHON IN ARBEIT!":PR
     INT:GOTO 54000                               <251>
7700 PRINT"NILS,DER ANDERE ZAUBERLEHRLING
     UND"                                         <170>
7710 PRINT"MEIKES FREUND,KOMM HEREIN UND E
     R-"                                          <104>
7720 PRINT"SCHLAEGT MICH AUS EIFERSUCHT!":
     PRINT:GOTO 54000                             <010>
```

**Listing 3. »The Sword« (Fortsetzung)**

```
7800 PRINT"REIZVOLLER GEDANKE...ABER DAS A
     BENTEUER"                                <152>
7810 PRINT"RUFT.ALSO BESTEHE ES UND SCHAFF
      DIR"                                    <017>
7820 PRINT"KEIN NEUES!":GOTO 1500             <172>
7900 PRINT"SIE REISST MIR DIE KLEIDER VOM
     LEIBE."                                  <114>
7910 PRINT"ICH STERBE AN MEINER ERREGUNG!"
     :PRINT:GOTO 54000                        <058>
8000 IF O2<>0 THEN PRINT"NICHT ALLES AUF E
     INMAL!":GOTO 1500                        <143>
8001 IF O(O1)<>SP AND NOT O(O1)THEN PRINT"
     SEH ICH NICHT,HAB ICH NICHT!":GOTO 15
     00                                       <020>
8002 ON O1 GOTO 8010,8010,8010,8010,8010,8
     010,8020,8015,8300,8300,8050,8060,807
     0                                        <073>
8003 ON O1-13 GOTO 8080,8090,8100,8110,830
     0,8300,8300,8300,8120,8130,8140          <097>
8004 ON O1-24 GOTO 8150,8160,8300,8300,814
     0,8300,8180,8170                         <245>
8010 PRINT"UNTERSUCH IHN DOCH SELBST!":GOT
     O 1500                                   <069>
8015 PRINT"KANNST DU NICHT AUF DEUTSCH RED
     EN!":GOTO 1500                           <127>
8020 PRINT"ICH BIN EIN SCHOENER,MUSKULOESE
     R"                                       <241>
8021 PRINT"JUENGLING.MEIN NAME IST NEVS EN
     RA.":GOTO 1500                           <231>
8050 PRINT"IST NICHT ZU SEHEN!":GOTO 1500     <140>
8060 PRINT"WIE ICH SCHON SAGTE:LIEBLICH!":
     GOTO 1500                                <210>
8070 PRINT"SIE IST WIRKLICH SEHR,SEHR NIED
     LICH!":GOTO 1500                         <253>
8080 PRINT"MAN,HAT DIE EIN TEMPERAMENT!":G
     OTO 1500                                 <237>
8090 PRINT"SIE BLEIBT COOL,UND SO SIEHT SI
     E AUCH":PRINT"AUS!":GOTO 1500            <186>
8100 PRINT"EIN NETTES MAEDEL":GOTO 1500       <058>
8110 PRINT"DIE HUETTE IST KLEIN UND SCHEIN
     T"                                       <006>
8111 PRINT"GEMUETLICH ZU SEIN.":GOTO 1500     <042>
8120 PRINT"ES IST VERGOLDET UND MIT EDELST
     EINEN":PRINT"BESETZT.":GOTO 1500         <205>
8130 PRINT"ICH SOLTE ES LESEN!":GOTO 1500     <226>
8140 PRINT"ES IST EIN EHER FREUNDLICHER ZW
     ERG!":GOTO 1500                          <026>
8150 PRINT"DAS IST ER...":GOTO 1500           <031>
8160 PRINT"DU INHALIERST ZUVIEL NEBEL...ES
      IST":PRINT"GIFTGAS...DU STIRBST!":PR
     INT                                      <050>
8161 GOTO 54000                               <091>
8170 PRINT"SIE KOENNTE NOCH NUETZLICH SEIN
     !":GOTO 1500                             <202>
8180 PRINT"ZAUBERHAFT!":GOTO 1500             <018>
8300 PRINT"SIEHT GANZ NORMAL AUS!":GOTO 15
     00                                       <225>
30000 ON V2 GOTO 31000,31500,32000,32500,3
      2500,33000,33500,34000,33500,34000,3
      4500                                    <051>
30050 ON V2-11 GOTO 35000,35010,35020,3503
      0,35040,35050,35500,35500,36000,3700
      0                                       <099>
30060 ON V2-21 GOTO 33500,34000,38000,3800
      0,39500,39700,40000                     <116>
30100 GOTO 1500                               <146>
31000 PRINT"ICH TRAGE:"                       <195>
31005 FL(3)=-1:FOR T=7 TO AO                  <175>
31010 IF O(T)=-1 THEN FL(3)=0:PRINT NO$(T)
      "."                                     <226>
31020 NEXT                                    <040>
31030 IF FL(3)THEN PRINT"GAR NICHTS!"         <246>
31040 GOTO 1500                               <070>
31500 IF SP=10 THEN PRINT"DIE MAEDELS SEHN
       ZUM ANBEISSEN AUS!":GOTO 1500          <110>
31510 IF SP=17 THEN PRINT"ICH BIN MUEDE!":
      GOTO 1500                               <166>
31520 IF SP=5 THEN PRINT"GEH HIER NICHT WE
      ITER!":GOTO 1500                        <081>
31530 IF SP=14 OR SP=15 THEN PRINT"MAN SIE
      HT DEN WALD VOR BAEUMEN NICHT!":GOTO
       1500                                   <129>
31540 IF SP=3 THEN PRINT"ICH WOLLT ICH WAE
      R EIN SEEBAER,AEH{6SPACE}-MANN...":G
      OTO 1500                                <039>
31550 IF SP=31 THEN PRINT"ZWERGE LIEBEN LI
      CHT!":GOTO 1500                         <078>
31555 IF SP=7 THEN PRINT"ICH SOLLTE DAS BO
      OT BESSER VERLASSEN!":GOTO 1500         <255>
31560 IF SP=12 THEN PRINT"GIB AUF,HIER KOM
      MST DU NICHT MEHR WEG!":GOTO 1500       <051>
31900 PRINT"HILF DIR SELBST,":ZU=ZU-1         <098>
31910 PRINT"DANN HILFT DIR GOTT!"             <110>
31920 GOTO 1500                               <188>
32000 PRINT"WILLST DU WIRKLICH AUFHOEREN?"    <028>
32010 GOSUB 50000:POKE 646,F2                 <132>
32020 IF LEFT$(X$,1)="J"THEN PRINT"NA GUT!
      ":GOTO 54000                            <054>
32030 PRINT"WARUM TIPPST DU DANN 'ENDE'?":
      GOTO 1500                               <131>
32500 GOTO 1080                               <181>
33000 ZU=ZU-1:PRINT"NICHT ZU DANKEN!":GOTO
       1500                                   <205>
33500 GOSUB 56500:PRINT"PLEASE WAIT!":ZU=Z
      U-1                                     <211>
33510 OPEN 15,8,15,"S:"+N$                    <057>
33520 CLOSE 15                                <083>
33530 OPEN 2,8,2,N$+",S,W"                    <212>
33540 PRINT#2,SP:PRINT#2,ZU:PRINT#2,RA$(32
      ):PRINT#2,NO$(25):PRINT#2,PO            <146>
33550 FOR T=1 TO AO:PRINT#2,O(T):NEXT         <019>
33560 FOR T=1 TO AF:PRINT#2,FL(T):NEXT        <225>
33600 CLOSE 2:PRINT"FERTIG."                  <096>
33700 GOTO 1500                               <191>
34000 GOSUB 56500:PRINT"PLEASE WAIT"          <144>
34030 OPEN 2,8,2,N$+",S,R"                    <199>
34040 INPUT#2,SP:INPUT#2,ZU:INPUT#2,RA$(32
      ):INPUT#2,NO$(25):INPUT#2,PO            <182>
34050 FOR T=1 TO AO:INPUT#2,O(T):NEXT         <187>
34060 FOR T=1 TO AF:INPUT#2,FL(T):NEXT        <137>
34100 CLOSE 2:PRINT"FERTIG."                  <088>
34200 GOTO 1080                               <104>
34500 PRINT"BILDSCHIRMFARBE{8SPACE}(F1+RET
      URN){2SPACE}< >":POKE 650,128           <084>
34510 GET A$:IF A$="{F1}"THEN F0=F0+1:IF F
      0=15 THEN F0=0                          <253>
34520 IF A$=CHR$(13)THEN 34540                <130>
34530 POKE 53280,F0:POKE 53281,F0:GOTO 345
      10                                      <134>
34540 PRINT"SCHRIFTFARBE EINGABE{3SPACE}(F
      1+RETURN){2SPACE}>{RVSON,SPACE,RVOFF
      }<{2LEFT}";                             <046>
34545 GET A$                                  <208>
34550 IF A$="{F1}"THEN F1=F1+1:IF F1=15 TH
      EN F1=0                                 <167>
34560 IF A$=CHR$(13)THEN 34580                <174>
34570 POKE 646,F1:PRINT"{RVSON,SPACE,RVOFF
      ,LEFT}";:GOTO 34545                     <100>
34580 POKE 646,F2:PRINT:PRINT"SCHRIFTFARBE
       AUSGABE{3SPACE}(F1+RETURN){2SPACE}>
      {RVSON,SPACE,RVOFF}<{2LEFT}";           <229>
34585 GET A$                                  <250>
34590 IF A$="{F1}"THEN F2=F2+1:IF F2=15 TH
      EN F2=0                                 <045>
34595 IF A$=CHR$(13)THEN 34600                <013>
34598 POKE 646,F2:PRINT"{RVSON,SPACE,RVOFF
      ,LEFT}";:GOTO 34585                     <130>
34600 PRINT:PRINT"SCHRIFTFARBE KOPFZEILE (
      F1+RETURN){2SPACE}>{RVSON,SPACE,RVOF
      F}<{2LEFT}";                            <137>
34610 GET A$                                  <019>
34620 IF A$="{F1}"THEN F3=F3+1:IF F3=15 TH
      EN F3=0                                 <167>
34630 IF A$=CHR$(13)THEN 34650                <178>
34640 POKE 646,F3:PRINT"{RVSON,SPACE,RVOFF
      ,LEFT}";:GOTO 34610                     <137>
34650 POKE 650,0:PRINT:GOTO 1080              <212>
35000 X$="N":GOTO 1520                        <065>
35010 X$="S":GOTO 1520                        <140>
35020 X$="W":GOTO 1520                        <151>
35030 X$="O":GOTO 1520                        <159>
35040 X$="OB":GOTO 1520                       <192>
35050 X$="U":GOTO 1520                        <053>
35500 PRINT"DU HASST";:ZU=ZU-1                <160>
35510 IF ZU=1 THEN PRINT" EINEN BEFEHL GEG
      EBEN!":GOTO 35525                       <102>
35511 IF ZU=0 THEN PRINT" NOCH KEINEN BEFE
      HL GEGEBEN!":GOTO 1500                  <187>
35520 PRINT ZU"BEFEHLE GEGEBEN!"              <033>
35525 IF PO=1 THEN PRINT"DU HASST EINEN PU
      NKT.":GOTO 35560                        <040>
35527 PRINT"DU HASST"PO"PUNKTE."              <149>
```

```
35560 PRINT"ABER DAS SPIEL HASST DU NOCH N
      ICHT "                                   <100>
35570 PRINT"GEWONNEN!":GOTO 1500              <011>
36000 X$=RIGHT$(X$,LEN(X$)-5)                  <080>
36010 IF LEFT$(X$,1)<>CHR$(34)THEN PRINT"A
      NFUEHRUNGSZEICHEN VERGESSEN!":GOTO 1
      500                                      <127>
36011 IF RIGHT$(X$,1)<>CHR$(34)THEN PRINT"
      WRONG INPUT!":GOTO 1500                  <113>
36020 FOR ZA=1 TO AZ                           <162>
36030 IF ZA$(ZA)=X$THEN 36310                  <055>
36040 NEXT                                     <235>
36300 PRINT X$",NICHTS PASSIERT!":GOTO 150
      0                                        <110>
36310 ON ZA GOTO 36500,36600,36700,36800       <044>
36500 IF SP<>41 THEN PRINT"HIER GIBT'S NIC
      HTS ZUM SCHWIMMEN!":GOTO 1500            <037>
36505 IF FL(9)THEN SP=18:FL(7)=0:PRINT"PUH
      ,BIN DOCH GERETTET!":GOTO 1080           <168>
36510 SP=18:FL(7)=0:FL(9)=-1:PRINT"PUH,BIN
      DOCH GERETTET!":PO=PO+1:GOTO 1080        <033>
36600 IF SP<>33 THEN PRINT"DAS GEHT LEIDER
      NICHT!":GOTO 1500                        <080>
36610 IF FL(5)=0 THEN PRINT"ICH HABE DAS S
      CHWERT NOCH NICHT!":GOTO 1500            <065>
36620 SP=2:PRINT"ZWOOOCHH...WAU,ECHT GEIL
      SO'N ZAUBER!":PO=PO+1:GOTO 1080          <213>
36700 IF SP<>33 THEN PRINT"ICH SEH DEN DRA
      CHEN JA GAR NICHT!":GOTO 1500            <041>
36710 IF FL(6)THEN PRINT"DER DRACHE IST SC
      HON TOT!":GOTO 1500                      <028>
36720 PRINT:PRINT:PRINT:PRINT"PENG,ZACK,BU
      FF..."                                   <233>
36730 PRINT"BLITZE ZUCKEN,FLAMMEN LODERN,D
      ER DRACHE"                               <086>
36731 PRINT"IST TOT,ENDLICH BESIEGT!"          <184>
36732 NO$(25)="DEN TOTEN DRACHEN":FL(6)=-1    <091>
36740 PRINT"{DOWN}DER HEILIGE MANN ERSCHEI
      NT.ER SAGT:"                             <168>
36750 PRINT CHR$(34);:POKE 646,F1              <000>
36760 PRINT"DU HASST MEINE ERWARTUNGEN VOL
      L"                                       <148>
36761 PRINT" ERFUELLT.ICH LEGE NUN MEIN AM
      T NIEDER"                                <244>
36762 PRINT" UND GEBE ALLES AN DICH,MEIN S
      CHUELER."                                <235>
36763 PRINT" LEB WOHL...";:POKE 646,F2:PRI
      NT CHR$(34)                              <113>
36770 PRINT"ICH RUFE IHM NOCH NACH,DOCH ER
      IST"                                     <044>
36771 PRINT"SCHON WEG.NUN BIN ICH DER ZAUB
      ERER UND"                                <137>
36772 PRINT"HEILIGE MANN."                     <225>
36773 PRINT"{DOWN}AN DER HAND ERSCHEINT EI
      NE SCHRIFT,DIE"                          <058>
36774 PRINT"BESAGT,DASS DAS ADVENTURE NOCH
      NICHT":PRINT"GESCHLAGEN IST!"            <219>
36776 PO=PO+1                                  <016>
36777 PRINT"{4DOWN}(MORE){2UP}"                <151>
36780 GET A$:IF A$=""THEN 36780                <040>
36785 FL(6)=-1:GOTO 1080                       <182>
36800 IF NOT O(32)THEN PRINT"HAB' KEINE LA
      MPE!":GOTO 1500                          <254>
36810 IF SP<>32 THEN PRINT"BIN DOCH NICHT
      BLIND!":GOTO 1500                        <148>
36820 FL(4)=-1:PRINT                           <218>
36830 PRINT"DER ZWERG FREUT SICH UEBER DAS
      LICHT."                                  <205>
36840 PRINT"ICH SCHENKE IHM DIE LAMPE."        <037>
36845 PRINT"ER VERSTECKT SIE SCHNELL IN SE
      INER"                                    <037>
36850 PRINT"HOEHLE.":O(32)=0                   <015>
36860 PRINT"ER BEGINNT ZU ERZAEHLEN:"          <173>
36861 PRINT CHR$(34);:POKE 646,F1              <111>
36862 PRINT"DER DRACHE IST MIT MAGIE EINFA
      CH ZU"                                   <064>
36863 PRINT" BESIEGEN...BENUTZE DEIN WISSE
      N!";                                     <084>
36864 POKE 646,F2:PRINT CHR$(34)               <058>
36865 PRINT"MEHR WILL ER NICHT SAGEN!"         <064>
36866 PRINT"ER VERSCHWINDET...":O(24)=0        <088>
36870 RA$(32)="IN EINER HELLERLEUCHTETEN{7
      SPACE}HOEHLE.":PO=PO+1:GOTO 1080         <196>
37000 IF SP<>17 THEN PRINT"BIN NICHT MUEDE
      !":GOTO 1500                             <178>
37010 PRINT"LANGSAM SINKE ICH IN DEN SCHLA
      F...":SP=18                              <248>
37020 FOR T=1 TO 2000:NEXT                     <065>
37030 PRINT"{DOWN}ICH ERWACHE,UND SEHE...H
      AS?"                                     <116>
37040 PRINT"ICH BIN JA GANZ WO ANDERS..."      <219>
37045 PO=PO+1:GOTO 1080                        <088>
38000 PRINT"EINE KOMPLETTE ANLEITUNG KANN
      ICH"                                     <243>
38010 PRINT"DIR SO AUF DIE SCHNELLE NICHT
      GEBEN,"                                  <140>
38020 PRINT"ABER EIN PAAR TRICKS:"             <021>
38030 PRINT"SAVE IMMER ALLES AB,BEVOR DU E
      TWAS GE-"                                <156>
38040 PRINT"FAEHRLICHES AUSPROBIERST."         <220>
38050 PRINT"UNTERSUCHE ALLES GENAU UND LES
      E ALLES"                                 <100>
38060 PRINT"SORGFAELLTIG."                     <149>
38070 PRINT"'SCORE' BZW. 'SPIELSTAND' SAGT
       DIR,WIE-"                               <051>
38080 PRINT"VIELE BEFEHLE DU SCHON GEGEBEN
       HASST,"                                 <022>
38090 PRINT"UND DEINE PUNKTZAHL."              <068>
38095 PRINT"ZAUBERSPRUECHE WERDEN ANGEWEND
      ET,INDEM"                                <019>
38100 PRINT"MAN 'SAGE "CHR$(34)"ZAUBERSPRU
      CH"CHR$(34)"' TIPPT."                    <041>
38110 PRINT"EIN ZAUBERSPRUCH IST EIN BEFEH
      L,DER"                                   <175>
38120 PRINT"RUECKWAERTS EINGEGEBEN WERDEN
      MUSS!":                                  <146>
38999 GOTO 1500                                <156>
39500 SYS 867:PRINT CHR$(14)                   <109>
39510 F0=0:F1=5:F2=3:F3=5                      <171>
39530 GOTO 39800                               <028>
39700 PRINT CHR$(142):SYS 820                  <190>
39710 F0=6:F1=0:F2=15:F3=15                    <159>
39800 :                                        <151>
39810 POKE 53280,F0:POKE 53281,F0              <043>
39830 GOTO 1080                                <146>
40000 PRINT"DIE ZEIT VERGEHT...":GOTO 1500     <123>
50000 X$="":A$="":POKE 646,F1:PRINT:PRINT"
      @{LEFT}";                                <125>
50030 GET A$:IF A$=""THEN 50030                <137>
50031 POKE 212,0                               <097>
50033 IF A$=CHR$(13)THEN 50300                 <178>
50034 IF A$="{HOME}"THEN GOSUB 50500:GOTO
      50030                                    <014>
50035 IF A$=CHR$(32)THEN 50080                 <248>
50036 IF A$=CHR$(34)THEN POKE 212,1:GOTO 5
      0080                                     <227>
50037 IF A$=CHR$(20)THEN GOSUB 50400:GOTO
      50030                                    <218>
50038 IF LEN(X$)>78 THEN 50030                 <221>
50050 IF A$<"A"OR A$>"Z"THEN 50030             <091>
50080 PRINT A$;"@{LEFT}";:X$=X$+A$:GOTO 50
      030                                      <199>
50300 IF LEN(X$)<1 THEN GOTO 50030             <168>
50301 PRINT" ":RETURN                          <137>
50400 L=LEN(X$):IF L<1 THEN RETURN             <086>
50415 X$=LEFT$(X$,L-1):PRINT"{SPACE,2LEFT}
      @{LEFT}";                                <243>
50430 RETURN                                   <193>
50500 IF LEN(X$)<1 THEN RETURN                 <108>
50501 PRINT"{SPACE,LEFT}";:FOR S=1 TO LEN(
      X$)                                      <055>
50510 PRINT"{SPACE,2LEFT}";:NEXT               <064>
50520 PRINT"@{LEFT}";:X$="":RETURN             <198>
51000 FOR I=1 TO 10:BE$(I)="":NEXT             <164>
51020 WZ=1                                     <091>
51030 FOR I=1 TO LEN(X$)                       <065>
51040 IF MID$(X$,I,1)=" "THEN GOSUB 51100:
      GOTO 51060                               <065>
51045 IF WZ>10 THEN PRINT"EINGABE IST ZU L
      ANG!":ER=1:RETURN                        <101>
51050 BE$(WZ)=BE$(WZ)+MID$(X$,I,1)             <083>
51060 NEXT:GOSUB 51100:WZ=WZ-1                 <124>
51070 RETURN                                   <071>
51100 REM AUSSORTIEREN                         <086>
51110 IC=0:FOR I1=1 TO AU                      <184>
51120 IF BE$(WZ)=AU$(I1)THEN IC=1              <084>
51130 NEXT                                     <083>
51140 IF IC=0 THEN WZ=WZ+1:RETURN              <180>
51150 BE$(WZ)="":RETURN                        <116>
52000 IF BE$(1)="SAGE"THEN V2=20:RETURN        <002>
52001 O2=0:V2=0:IF WZ>1 THEN 52008             <164>
52002 FOR T=1 TO A2:IF V2$(T)=BE$(1)THEN V
      2=T:RETURN                               <163>
52003 NEXT                                     <196>
```

```
52008 IF WZ>3 THEN PRINT"WAS SOLL DAS BEDE
      UTEN?":GOTO 1500                       <218>
52009 FOR T=1 TO AV                          <216>
52010 IF VE$(T)=BE$(1)THEN V=T:GOTO 52100    <223>
52020 NEXT                                   <213>
52030 PRINT"'"BE$(1)"' KENNE ICH NICHT!":E
      R=1:RETURN                             <108>
52100 FOR T=1 TO AO                          <044>
52110 IF O$(T)=BE$(2)THEN O1=T:GOTO 52150    <192>
52120 NEXT                                   <057>
52130 PRINT"KEIN NOMEN GEFUNDEN!":ER=1:RET
      URN                                    <033>
52150 IF WZ=2 THEN RETURN                    <029>
52200 FOR T=1 TO AO                          <144>
52210 IF O$(T)=BE$(3)THEN O2=T:GOTO 52300    <062>
52220 NEXT                                   <157>
52230 PRINT"'"BE$(3)"' KENNE ICH NICHT!":E
      R=1                                    <226>
52300 RETURN                                 <031>
53000 X=PEEK(211):Y=PEEK(214):POKE 646,F3    <198>
53010 PRINT"{HOME,RVSON,31SPACE}THE SWORD{
      RVOFF}"                                <025>
53020 PRINT"{HOME,RVSON}";KR$(SP);"{RVOFF}
      "                                      <127>
53030 POKE 211,X:POKE 214,Y:SYS 58732        <051>
53050 RETURN                                 <019>
54000 POKE 646,F2:PRINT"NOCHMAL?"            <220>
54010 GOSUB 50000:IF LEFT$(X$,1)="J"THEN P
      OKE 53280,6:POKE 53281,6:RUN 20        <246>
54020 GOTO 60000                             <175>
56000 SYS 867:RETURN                         <206>
56500 PRINT"FILE-NUMMER (1-3) ⊞{LEFT}";      <157>
56510 GET E$:IF E$<>"1"AND E$<>"2"AND E$<>
      "3"THEN 56510                          <017>
56511 IF E$="1"THEN N$="C"                   <163>
56512 IF E$="2"THEN N$="D"                   <102>
56513 IF E$="3"THEN N$="E"                   <039>
56520 PRINT E$"←":RETURN                     <214>
57000 POKE 646,F2:PRINT                      <183>
57010 PRINT"DER HEILIGE MANN ERBLICKT SEIN
      EN"                                    <165>
57020 PRINT"SCHUELER (MICH) UND SAGT:"       <069>
57030 PRINT CHR$(34);:POKE 646,F1            <214>
57040 PRINT"ICH BIN STOLZ,MEIN SCHUELER!"    <227>
57050 PRINT" DU WOLLTEST ZAUBERER WERDEN,U
      ND JETZT"                              <139>
57060 PRINT" STEHST DU VOR DER ABSCHLIESSE
      NDEN"                                  <247>
57070 PRINT" PRUEFUNG.TOETE DEN DRACHEN,DE
      R IN DEM"                              <227>
57080 PRINT" VULKAN 'VETUS' LEBT UND NIMM
      DAS"                                   <096>
57085 PRINT" ZAUBERSCHWERT,DAS ER BEWACHT!
      "                                      <134>
57090 PRINT"{SHIFT-SPACE}GELINGT DIR DIES,
      BIST DU WAHRLICH"                      <121>
57091 PRINT" FUER WUERDIG BEFUNDEN,EIN ZAU
      BERER"                                 <001>
57092 PRINT" ZU SEIN.";:POKE 646,F2:PRINT
      CHR$(34)                               <131>
57093 PRINT"DER HEILIGE MANN,MEIN LEHRMEIS
      TER,"                                  <143>
57094 PRINT"SCHEINT SICH IN LUFT AUFZULOES
      EN!"                                   <172>
57095 PRINT"JETZT BIN ICH AUF MICH GESTELL
      T!"                                    <025>
57100 PRINT"{2DOWN}<MORE>{2UP}"              <054>
57110 GET A$:IF A$=""THEN 57110              <168>
57120 RETURN                                 <252>
60000 POKE 59639,16:POKE 53280,6:POKE 5328
      1,6                                    <214>
60010 SYS 820                                <163>
60020 PRINT"{4DOWN,12SPACE,LIG.RED,2SPACE}
      PINK ARTS:"                            <134>
60030 PRINT"{2DOWN,8SPACE,BLACK}£{2SPACE}T
       H E{2SPACE}S W O R D{2SPACE}£"        <149>
60031 PRINT"{2DOWN,5SPACE}@ 1986 BY A.SOMM
      ER AND S.ADOMAT"                       <085>
60035 PRINT"{5DOWN}************************
      *****************"                     <025>
60040 M$="{8SPACE}THE SWORD{SPACE,WHITE}BY
      {SPACE,LIG.RED}PINK ARTS{BLACK}":GOS
      UB 61000                               <092>
60050 M$="{7SPACE}PROGRAMING : ARNE SOMMER
      ":GOSUB 61000                          <165>
60060 M$="{6SPACE}IDEA : A.SOMMER AND S.AD
      OMAT":GOSUB 61000                      <119>
60065 M$="{2SPACE}STORY AND MAP : A.SOMMER
       + S.ADOMAT":GOSUB 61000               <148>
60070 M$="{5SPACE}SPECIAL EFFECTS : SVEN A
      DOMAT":GOSUB 61000                     <217>
60080 M$="{5SPACE}GRAFICS : A.SOMMER + S.A
      DOMAT":GOSUB 61000                     <253>
60090 M$="{5SPACE}MOTIVATION : BIRTHE NIET
      MANN":GOSUB 61000                      <117>
60100 M$="{5SPACE}GREETINGS TO ALL OUR FRI
      ENDS!":GOSUB 61000                     <162>
60110 M$="AND COMING SOON : RITTER + SWORD
       PART 2{LEFT}":GOSUB 61000             <025>
60120 M$=" AEHNLICHKEITEN MIT WIRKLICHEN B
      EGEBEN-{2SPACE}HEITEN UND NAMEN SIND
      REIN"                                  <237>
60130 M$=M$+" ZUFAELLIG!":GOSUB 61000        <198>
60900 GOTO 60040                             <196>
61000 POKE 214,24:POKE 211,0:SYS 58732       <053>
61010 PRINT M$:PRINT:PRINT:GOSUB 62000       <076>
61020 PRINT:GOSUB 62000:PRINT:GOSUB 62000:
      PRINT:GOSUB 62000:PRINT:GOTO 62000     <072>
61030 RETURN                                 <125>
62000 FOR T=1 TO 600:NEXT:RETURN             <082>
```

**Listing 3. »The Sword« (Schluß)**

# Freiheit

**Warum mußten Sie auch dem König von Folteranien unangenehm auffallen? Nun müssen Sie einen Ausweg aus seinem Kerker finden.**

Freiheit ist ein deutschsprachiges Textadventure, das aus 28 KByte reiner Maschinensprache besteht. Die Aufgabe des Spieles ist ein Gefängnisausbruch von einer Insel. Nach vielen Problemen auf der Nachbarinsel endet die Suche, wenn Sie endlich im Flugzeug nach Hause sitzen. Zum Spiel selbst: Es existieren beinahe 100 Räume, in denen sich etwa zwei Dutzend Personen aufhalten können. Außerdem gibt es natürlich viele diverse Gegenstände sowie einen knapp 150 Worte umfassenden Sprachschatz, der sich mit dem Befehl »VERBEN« ausgeben läßt.

Alle Eingaben sind voll abgesichert. Es werden ganze Sätze verstanden. Bei Eingaben, die sich auf mehrere Dinge beziehen, sind diese ALLE anzugeben. Ein Beispiel: »NIMM DIE MÜNZE AUS DER KISTE« ist gleichbedeutend zu »NIMM MÜNZE KISTE« oder »NIMM KISTE MÜNZE«. Das Wort »UND« ist zulässig, hat aber eine Einschränkung: will man zum Beispiel die Münze und den Stock in die Kiste legen, kann »UND« nicht verwendet werden, da folgender Satz zweideutig wäre:

»LEGE DIE MÜNZE UND DEN STOCK IN DIE KISTE«.

Der Computer nimmt an, Sie wollen die Münze auf den Boden und den Stock in die Kiste legen. Dies ist aber ein seltener Effekt. Die Eingabe von Abkürzungen ist bis auf einige Vordefinitionen der wichtigsten Verben nicht zugelassen. Für die Eingabe bleibt dem Spieler nur eine Minute, danach wird »WARTE« ausgeführt, und die eigene Eingabe geht verloren. »LADE« und »SPEICHERE« bedürfen keiner weiteren Erläuterung. Man wird durch ein kleines Menü geführt (Kassette oder Diskette).

Am Schluß noch ein Tip: speichern Sie gelegentlich den Zwischenspielstand. Es könnte plötzlich eine Person auftauchen, die einen möglicherweise sucht...

(Bodo Mertins/dm)

**Listing »Freiheit«. Bitte mit dem MSE eingeben**

```
PROGRAMM : FREIHEIT     0801 7807
----------------------------------
0801 : 0B 08 FF FF 9E 32 30 37  BB
0809 : 30 00 00 00 00 00 00 00  3A
0811 : 00 00 00 1F 08 78 20 A3  01
0819 : FD 20 15 FD 20 5B FF EA  DE
0821 : EA EA 5B 4C 64 29 20 D6  DE
0829 : 09 0D CB 4F 4D 4D 41 4E  76
0831 : 44 4F 3F 20 A4 9D 00 A9  7B
0839 : 00 8D D5 09 8D E3 08 85  B9
0841 : A0 85 A1 85 A2 85 C6 20  6F
0849 : E4 FF 85 02 A5 A1 C9 0E  79
0851 : 90 03 4C BE 08 A5 02 F0  E5
0859 : EE C9 5B B0 EA C9 41 B0  7C
0861 : 13 C9 22 F0 12 C9 20 F0  D1
0869 : 0E C9 14 F0 28 C9 0D F0  66
0871 : 3B 4C 48 08 18 69 80 AC  0E
0879 : D5 09 C0 E3 B0 14 C8 8C  67
0881 : D5 09 99 E2 08 20 D2 FF  6A
0889 : A9 00 85 D4 20 D6 09 A4  54
0891 : 9D 00 4C 48 08 AC D5 09  9A
0899 : C0 00 F0 F6 88 8C D5 09  CB
08A1 : 20 D6 09 9D A4 20 9D 9D  1F
08A9 : 00 4C 48 08 AC D5 09 F0  62
08B1 : 96 A9 00 99 E3 08 20 D6  FC
08B9 : 09 20 0D 00 60 20 D6 09  8A
08C1 : 20 0D 00 A9 D7 8D E3 08  27
08C9 : A9 C1 8D E4 08 A9 D2 8D  87
08D1 : E5 08 A9 D4 8D E6 08 A9  43
08D9 : C5 8D E7 08 A9 00 8D E8  02
08E1 : 08 60 D3 D4 CF D0 00 00  2C
08E9 : C5 D2 C5 00 D5 C6 C5 D2  D9
08F1 : 20 C7 CF CC C4 00 C9 D3  9D
08F9 : D4 C5 00 C1 D3 D3 20 CD  E0
0901 : C9 D2 00 D3 D3 00 00 D3  93
0909 : C1 C6 D4 00 D3 C5 CC 20  41
0911 : D5 CE C4 20 C7 C5 C8 20  90
0919 : C2 CF CF D4 20 D5 CE C4  C7
0921 : 20 CE 20 D5 CE C4 20 CE  9C
0929 : 00 CE C4 20 C8 20 D5 CE  48
0931 : C4 20 CF 00 D2 00 00 BE  43
0939 : 31 36 31 00 46 09 00 00  7E
0941 : 91 23 31 34 00 50 09 00  DD
0949 : 00 B3 47 45 54 31 00 5C  25
0951 : 09 00 00 87 49 4E 57 41  32
0959 : 49 54 00 67 09 00 00 47  D8
0961 : 45 54 31 BE 32 00 70 09  E5
0969 : 00 00 B6 47 45 54 00 79  EA
0971 : 09 00 00 91 23 39 31 00  6D
0979 : 82 09 00 00 B4 47 45 54  C3
0981 : 00 8B 09 00 00 91 23 36  0F
0989 : 35 00 97 09 00 00 B4 49  2B
0991 : 4E 50 55 54 34 00 A0 09  BF
0999 : 00 00 91 23 33 34 00 AC  90
09A1 : 09 00 00 B6 49 4E 50 55  74
09A9 : 54 31 00 B5 09 00 00 91  00
09B1 : 23 33 32 00 C1 09 00 00  5F
09B9 : B6 49 4E 50 55 54 31 00  6E
09C1 : CA 09 00 00 91 23 32 30  6B
09C9 : 00 D6 09 00 00 B6 49 4E  EE
09D1 : 50 55 54 32 04 68 85 FB  B9
09D9 : 68 85 FC A0 00 4C E8 09  6F
09E1 : B1 FB F0 0C 20 D2 FF E6  B4
09E9 : FB D0 F5 E6 FC 4C E1 09  72
09F1 : E6 FB D0 02 E6 FC 6C FB  49
09F9 : 00 00 11 00 00 00 00 00  3E
0A01 : 00 00 00 00 02 05 03 83  5D
0A09 : 85 04 06 02 04 06 02 06  D7
0A11 : 02 06 02 06 02 06 02 00  B0
0A19 : 23 0A 00 00 90 23 32 32  91
0A21 : 37 00 30 0A 00 00 B4 49  0B
0A29 : 4E 50 31 45 4E 44 00 36  08
0A31 : 0A 00 00 A9 00 3F 0A 00  93
0A39 : 00 8A 42 45 50 00 4E 0A  0A
0A41 : 00 00 8B 42 45 53 54 52  51
0A49 : BF 31 2C 59 00 59 0A 00  CA
0A51 : 00 B7 BB 46 46 44 32 00  34
0A59 : 61 0A 00 00 8E 23 30 00  82
0A61 : 6A 0A 00 00 BB 32 31 32  44
0A69 : 00 76 0A 00 00 B7 4F 55  CD
0A71 : 54 50 55 54 00 82 0A 00  09
0A79 : 00 B9 22 A4 9D 22 2C 30  6F
0A81 : 00 92 0A 00 11 0A 07 01  CC
0A89 : 01 04 01 0B 0C 01 01 01  FD
0A91 : 01 01 01 01 01 01 01 50  30
0A99 : 55 54 32 8D 42 45 50 00  E6
0AA1 : A9 0A 00 00 90 23 30 00  32
0AA9 : B6 0A 00 00 B6 49 4E 50  F4
0AB1 : 31 45 4E 44 00 BC 0A 00  AF
0AB9 : 00 A1 00 C5 0A 00 00 8A  F8
0AC1 : 42 45 50 00 D1 0A 00 00  27
0AC9 : B7 4F 55 54 50 55 54 00  09
0AD1 : E0 0A 00 00 B9 22 9D A4  23
0AD9 : 20 9D 9D 22 2C 30 00 E9  8C
0AE1 : 0A 00 00 87 47 45 54 00  CC
0AE9 : F8 0A 00 00 49 4E 50 55  D9
0AF1 : 54 33 8D 42 45 50 00 01  63
0AF9 : 0B 00 00 B6 47 45 54 00  CB
0B01 : 09 0B 00 00 01 01 00 00  A8
0B09 : 00 03 06 01 D5 CE C4 03  19
0B11 : 01 01 C4 C5 D2 03 01 02  CA
0B19 : C4 C9 C5 03 01 03 C4 C1  52
0B21 : D3 03 01 04 C4 C5 CE 03  F2
0B29 : 01 05 C4 C5 CD 05 01 06  AC
0B31 : D5 C5 C2 C5 D2 05 01 07  B9
0B39 : D5 CE D4 C5 D2 03 01 08  BC
0B41 : C1 D5 C6 03 01 09 D6 CF  52
0B49 : CE 03 01 0A D6 CF CD 02  42
0B51 : 01 0B C9 CE 02 01 0C C9  10
0B59 : CD 02 01 0D C1 CD 02 01  9E
0B61 : 0E C1 CE 03 05 02 C5 C4  65
0B69 : C5 0D 05 02 CD C9 D4 C7  44
0B71 : C5 C6 C1 CE C7 C5 CE C5  55
0B79 : D2 06 05 02 C6 D2 C5 D5  96
0B81 : CE C4 07 05 08 D7 C1 C5  E6
0B89 : D2 D4 C5 D2 07 05 09 CB  E5
0B91 : C5 CC CC CE C5 D2 04 05  D6
0B99 : 09 CF C2 C5 D2 04 05 09  67
0BA1 : CB CF C3 C8 11 05 0A CB  57
0BA9 : CC C5 C9 C4 C5 D2 D6 C5  3D
0BB1 : D2 CB C1 C5 D5 C6 C5 D2  E2
0BB9 : 0A 05 0A D6 C5 D2 CB C1  49
0BC1 : C5 D5 C6 C5 D2 09 05 0B  7B
0BC9 : C6 C1 C5 CC D3 C3 C8 C5  85
0BD1 : D2 0D 05 0B D0 C1 D3 D3  DF
0BD9 : C6 C1 C5 CC D3 C3 C8 C5  95
0BE1 : D2 05 05 0C C2 C1 D5 C5  16
0BE9 : D2 08 05 0C CC C1 CE C4  22
0BF1 : D7 C9 D2 D4 07 05 0D CD  64
0BF9 : CF CE D4 C5 D5 D2 08 05  3B
0C01 : 0F D0 CF CC C9 DA C9 D3  48
0C09 : D4 07 05 11 D3 D4 D2 C5  7F
0C11 : C9 C6 C5 08 05 12 C6 CF  4B
0C19 : C5 D2 D3 D4 C5 D2 0A 05  FC
0C21 : 13 D3 D4 C5 D7 C1 D2 C4  6C
0C29 : C5 D3 D3 12 05 14 D4 C9  E7
0C31 : C3 CB C5 D4 D6 C5 D2 CB  64
0C39 : C1 C5 D5 C6 C5 D2 C9 CE  E3
0C41 : 0C 05 14 D6 C5 D2 CB C1  55
0C49 : C5 D5 C6 C5 D2 C9 CE 11  3C
0C51 : 05 16 C6 CC D5 C7 C8 C1  EF
0C59 : C6 C5 CE D0 CF CC C9 DA  10
0C61 : C9 D3 D4 08 05 17 DA CF  5E
0C69 : C5 CC CC CE C5 D2 07 05  BA
0C71 : 18 D0 C1 D3 D3 C1 CE D4  0C
0C79 : 03 05 01 CD C9 D2 06 05  4E
0C81 : 01 C7 D2 C5 C7 CF D2 05  23
0C89 : 04 01 CB C9 D3 D4 C5 07  43
0C91 : 04 02 C2 CF D4 D4 C9 C3  E3
0C99 : C8 06 04 03 CB C1 D3 D4  8A
0CA1 : C5 CE 07 04 04 D3 C3 C8  8F
0CA9 : D2 C1 CE CB 07 04 05 C2  B3
0CB1 : D2 D5 CE CE C5 CE 09 04  FA
0CB9 : 06 C2 C5 C8 C1 C5 CC D4  D2
0CC1 : C5 D2 04 04 07 D2 CF C8  49
0CC9 : D2 09 04 08 C3 CF CE D4  C2
0CD1 : C1 C9 CE C5 D2 0B 04 09  8B
0CD9 : C1 CC C1 D2 CD C1 CE CC  8B
0CE1 : C1 C7 C5 0A 04 0A CD D5  AC
0CE9 : C5 CC CC D4 CF CE CE C5  1C
0CF1 : 05 04 0B D4 CF CE CE C5  90
0CF9 : 04 03 01 D0 C1 D3 D3 03  E9
0D01 : 03 02 D5 C8 D2 06 03 03  03
0D09 : D4 C9 C3 CB C5 D4 04 03  45
0D11 : 04 C7 C5 CC C4 0A 03 05  B6
0D19 : C8 D5 D3 D4 C5 CE D3 C1  01
0D21 : C6 D4 05 03 06 C1 CE DA  52
0D29 : D5 C7 04 03 07 C7 CF CC  CB
0D31 : C4 0A 03 08 D3 C3 C8 CC  D4
0D39 : D5 C5 D3 D3 C5 CC 05 03  3D
0D41 : 09 C5 D3 D3 C5 CE 0F 03  B1
0D49 : 0A DA D5 C5 CE C4 D3 C3  D8
0D51 : C8 CC D5 C5 D3 D3 C5 CC  3A
0D59 : 03 03 0B C1 D8 D4 04 03  23
0D61 : 0C D3 C5 C9 CC 07 03 0D  2C
0D69 : D3 C3 C8 CD D5 C3 CB 06  C0
0D71 : 03 0E C2 C5 C3 C8 C5 D2  24
0D79 : 09 03 0F C2 CC C5 C9 D3  EA
0D81 : D4 C9 C6 D4 04 03 10 C2  A4
0D89 : C1 CC CC 07 03 11 DA C5  74
0D91 : C9 D4 D5 CE C7 06 03 12  F0
0D99 : C2 D2 C9 CC CC C5 04 03  E1
0DA1 : 13 C4 CF D3 C5 09 03 14  5D
0DA9 : CB CC C5 C5 C2 CC C1 D4  48
0DB1 : D4 0A 03 15 C5 D3 D3 C5  C4
0DB9 : CE CD C1 D2 CB C5 04 01  35
0DC1 : 01 CE C1 C3 C8 08 01 02  E7
0DC9 : D2 C9 C3 C8 D4 D5 CE C7  50
0DD1 : 01 02 01 CE 04 02 01 CE  DF
0DD9 : CF D2 C4 06 02 01 CE CF  06
0DE1 : D2 C4 C5 CE 01 02 02 D3  30
0DE9 : 04 02 02 D3 D5 C5 C4 06  94
0DF1 : 02 02 D3 D5 C5 C4 C5 CE  DB
0DF9 : 01 02 03 D7 04 02 03 D7  C3
0E01 : C5 D3 D4 06 02 03 D7 C5  C9
0E09 : D3 D4 C5 CE 01 02 04 CF  61
0E11 : 03 02 04 CF D3 D4 05 02  0C
0E19 : 04 CF D3 D4 C5 CE 02 02  73
0E21 : 05 CE D7 08 02 05 CE CF  A7
```

**Listing »Freiheit« (Fortsetzung)**
**Bitte mit dem MSE eingeben**

```
0E29 : D2 C4 D7 C5 D3 D4 0A 02   1C
0E31 : 05 CE CF D2 C4 D7 C5 D3   B5
0E39 : D4 C5 CE 02 02 06 CE CF   0F
0E41 : 07 02 06 CE CF D2 C4 CF   EB
0E49 : D3 D4 09 02 06 CE CF D2   C5
0E51 : C4 CF D3 D4 C5 CE 02 02   6B
0E59 : 07 D3 D7 08 02 07 D3 D5   94
0E61 : C5 C4 D7 C5 D3 D4 0A 02   47
0E69 : 07 D3 D5 C5 C4 D7 C5 D3   52
0E71 : D4 C5 CE 02 02 08 D3 CF   6B
0E79 : 07 02 08 D3 D5 C5 C4 CF   3C
0E81 : D3 D4 09 02 08 D3 D5 C5   43
0E89 : C4 CF D3 D4 C5 CE 01 02   9F
0E91 : 09 C8 04 02 09 C8 CF C3   DD
0E99 : C8 04 02 09 D2 C1 D5 C6   25
0EA1 : 04 02 09 CF C2 C5 CE 01   7A
0EA9 : 02 0A D2 06 02 0A D2 D5   8D
0EB1 : CE D4 C5 D2 01 02 0A D5   A9
0EB9 : 05 02 0A D5 CE D4 C5 CE   45
0EC1 : 01 02 0B C7 04 02 0B C7   8B
0EC9 : C5 C8 C5 05 02 0B D2 C5   54
0ED1 : CE CE C5 04 02 0B D2 C5   48
0ED9 : CE CE 03 02 0B C7 C5 C8   A7
0EE1 : 04 04 0C C2 CF CF D4 0B   28
0EE9 : 04 0D D3 D4 C5 C9 CE D0   8B
0EF1 : CC C1 D4 D4 C5 05 02 0C   12
0EF9 : C8 C9 CC C6 C5 08 02 0D   70
0F01 : C9 CE D6 C5 CE D4 D5 D2   30
0F09 : 01 02 0D C9 04 02 0E CC   AA
0F11 : C1 C4 C5 09 02 0F D3 D0   50
0F19 : C5 C9 C3 C8 C5 D2 C5 05   E1
0F21 : 02 10 D0 C1 D5 D3 C5 04   B3
0F29 : 02 11 C5 CE C4 C5 04 02   8D
0F31 : 11 D3 D4 CF D0 05 02 12   BC
0F39 : D7 C1 D2 D4 C5 06 02 13   FB
0F41 : D6 C5 D2 C2 C5 CE 07 02   FA
0F49 : 13 C2 C5 C6 C5 C8 CC C5   69
0F51 : 03 02 14 D3 C1 C7 04 02   43
0F59 : 14 D3 C1 C7 C5 04 02 14   6D
0F61 : D2 C5 C4 C5 06 02 14 D3   68
0F69 : C3 C8 D2 C5 C9 07 02 14   03
0F71 : D3 C3 C8 D2 C5 C9 C5 08   84
0F79 : 02 14 C1 CE D4 D7 CF D2   C0
0F81 : D4 C5 05 02 15 D3 C3 C8   4A
0F89 : C1 D5 06 02 15 D3 C3 C8   87
0F91 : C1 D5 C5 0A 02 15 D5 CE   AD
0F99 : D4 C5 D2 D3 D5 C3 C8 C5   A9
0FA1 : 05 02 16 C8 CF D2 C3 C8   7A
0FA9 : 06 02 16 C8 CF D2 C3 C8   83
0FB1 : C5 05 02 16 C8 CF C5 D2   04
0FB9 : C5 09 02 17 DA C5 D2 D3   35
0FC1 : D4 CF C5 D2 C5 08 02 17   1B
0FC9 : DA C5 D2 C8 C1 C3 CB C5   48
0FD1 : 09 02 18 D3 C3 C8 CC C9   A5
0FD9 : C5 D3 D3 C5 08 02 18 D3   CE
0FE1 : C3 C8 CC C9 C5 D3 D3 08   CF
0FE9 : 02 19 D6 C5 D2 CC C9 C5   2C
0FF1 : D2 C5 07 02 19 D6 C5 D2   AD
0FF9 : CC C9 C5 D2 04 02 19 CC   C4
1001 : C5 C7 C5 04 02 19 D7 C9   78
1009 : D2 C6 05 02 19 D7 C5 D2   CD
1011 : C6 C5 03 02 1A C9 D3 D3   A2
1019 : 04 02 1A C5 D3 D3 C5 03   57
1021 : 02 1B C7 C9 C2 04 02 1B   66
1029 : C7 C5 C2 C5 06 02 1C CF   BD
1031 : C5 C6 C6 CE C5 04 02 1D   A3
1039 : CE C9 CD CD 05 02 1D CE   8B
1041 : C5 C8 CD C5 04 02 1D C8   ED
1049 : CF CC C5 03 02 1D C8 CF   1C
1051 : CC 05 02 1D D4 D2 C1 C7   3F
1059 : C5 04 02 1E C8 C5 C2 C5   B6
1061 : 03 02 1E C8 C5 C2 03 01   B6
1069 : 01 C9 CE D3 08 02 1F D6   38
1071 : C5 D2 CC C1 D3 D3 C5 06   0A
1079 : 05 0C C2 C1 D5 C5 D2 CE   E1
1081 : 03 01 01 C1 D5 D3 03 01   87
1089 : 01 CD C9 D4 09 04 0C CD   FA
1091 : CF D4 CF D2 C2 CF CF D4   AC
1099 : 00 00 00 00 00 B9 31 32   90
10A1 : 2C 35 2C 31 32 2C 22 D6   54
10A9 : C5 D2 CB C1 C5 D5 C6 C5   B4
10B1 : D2 C9 CE 22 00 D7 10 00   5F
10B9 : 00 B9 31 37 2C 35 2C 31   49
10C1 : 33 2C 22 C6 CC D5 C7 C8   98
10C9 : C1 C6 C5 CE D0 CF CC C9   8B
10D1 : DA C9 D3 D4 22 00 EE 10   1D
10D9 : 00 00 B9 38 2C 35 2C 31   CE
10E1 : 34 2C 22 DA CF C5 CC CC   07
10E9 : CE C5 D2 22 00 04 11 00   F7
10F1 : 00 B9 37 2C 35 2C 31 35   05
10F9 : 2C 22 D0 C1 D3 D3 C1 CE   23
1101 : D4 22 00 16 11 00 00 B9   2E
1109 : 33 2C 35 2C 31 36 2C 22   DF
1111 : CD C9 D2 22 00 2B 11 00   59
1119 : 00 B9 36 2C 35 2C 31 36   EF
1121 : 2C 22 C7 D2 C5 C7 CF D2   2A
1129 : 22 00 3E 11 00 00 B9 35   4E
1131 : 2C 34 2C 31 2C 22 CB C9   3F
1139 : D3 D4 C5 22 00 53 11 00   0B
1141 : 00 B9 37 2C 34 2C 32 2C   37
1149 : 22 C2 CF D4 D4 C9 C3 CB   97
1151 : 22 00 67 11 00 00 B9 36   C3
1159 : 2C 34 2C 33 2C 22 CB C1   97
1161 : D3 D4 C5 CE 22 00 7C 11   20
1169 : 00 00 B9 37 2C 34 2C 4C   6C
1171 : 9E 12 AD FA 09 D0 F8 20   1E
1179 : 27 08 A9 08 85 58 A9 E3   99
1181 : 85 57 A9 00 85 59 BD 05   80
1189 : 0B 8D 07 0B 8D 08 0B A0   05
1191 : FF AD 08 0B F0 0D AD 07   07
1199 : 0B 49 80 8D 07 0B A9 00   8A
11A1 : 8D 08 0B C8 B1 57 F0 C7   37
11A9 : C9 C1 B0 0C C9 20 F0 E1   26
11B1 : A9 80 8D 08 0B 4C 92 11   7E
11B9 : 18 98 65 57 85 57 A9 00   1B
11C1 : 65 58 85 58 A0 FF 84 59   8D
11C9 : C8 B1 57 F0 04 C9 C1 B0   55
11D1 : F7 84 59 20 3A 12 20 55   C4
11D9 : 12 B0 2D AD 53 12 C9 01   33
11E1 : F0 16 AC 05 0B C8 8C 05   DB
11E9 : 0B AD 53 12 0D 07 0B 99   4A
11F1 : 04 0A AD 54 12 99 84 0A   04
11F9 : 18 A5 57 65 59 85 57 A5   D1
1201 : 58 69 00 85 58 4C 90 11   0B
1209 : 20 D6 09 C9 43 48 20 4B   9D
1211 : 45 4E 4E 45 20 27 00 A6   42
1219 : 59 A0 00 B1 57 20 D2 FF   BA
1221 : C8 CA D0 F7 20 D6 09 27   AD
1229 : 20 4E 49 43 48 54 21 0D   F1
1231 : 00 A9 00 8D FA 09 4C 73   C8
1239 : 11 A9 0B 85 5B A9 0A 85   C9
1241 : 5A 60 84 5C A0 00 B1 5A   FD
1249 : E6 5A D0 02 E6 5B A4 5C   65
1251 : 60 04 02 11 20 43 12 8D   D6
1259 : 52 12 20 43 12 8D 53 12   24
1261 : 20 43 12 8D 54 12 AD 52   8A
1269 : 12 D0 03 4C 9C 12 C5 59   52
1271 : F0 0C AE 52 12 20 43 12   B1
1279 : CA D0 FA 4C 55 12 A9 00   80
1281 : 8D 09 0B A8 A6 59 20 43   A7
1289 : 12 D1 57 F0 03 EE 09 0B   5A
1291 : C8 CA D0 F2 AD 09 0B D0   42
1299 : BB 18 60 38 60 AD FA 09   F1
12A1 : D0 19 A9 00 8D FB 09 8D   60
12A9 : FE 09 8D FF 09 8D 00 0A   A0
12B1 : 8D 03 0A 8D 04 0A A9 00   2B
12B9 : 8D 06 0B A9 00 8D FC 09   B4
12C1 : 8D FD 09 8D 01 0A 8D 02   DB
12C9 : 0A 8D FA 09 AD 06 0B CD   4C
12D1 : 05 0B D0 03 4C C6 13 AC   91
12D9 : 06 0B C8 8C 06 0B B9 04   D0
12E1 : 0A C9 02 F0 2D C9 03 F0   7E
12E9 : 4F C9 04 F0 66 C9 05 F0   E7
12F1 : 70 C9 82 F0 7A C9 83 D0   AA
12F9 : 03 4C B9 13 C9 84 D0 03   F1
1301 : 4C A6 13 C9 85 D0 03 4C   22
1309 : B6 13 A9 01 8D FA 09 4C   41
1311 : C6 13 B9 84 0A 8D FB 09   6F
1319 : A9 00 8D FC 09 8D FD 09   CC
1321 : 8D FE 09 8D FF 09 8D 00   A0
1329 : 0A 8D 01 0A 8D 02 0A 8D   A8
1331 : 03 0A 8D 04 0A 4C CD 12   7C
1339 : B9 84 0A AE FC 09 D0 03   EE
1341 : 8D FC 09 8D FD 09 CD FC   9A
1349 : 09 D0 05 A9 00 8D FD 09   A7
1351 : 4C CD 12 B9 84 0A 8D FE   0C
1359 : 09 A9 00 8D FF 09 4C CD   FE
1361 : 12 B9 84 0A 8D FF 09 A9   02
1369 : 00 8D FE 09 4C CD 12 B9   00
1371 : 84 0A 29 7F 8D 00 0A A9   89
1379 : 00 8D 01 0A 8D 02 0A 8D   EE
1381 : 03 0A 8D 04 0A 4C CD 12   CC
1389 : B9 84 0A 29 7F AE 01 0A   B2
1391 : D0 03 8D 01 0A 8D 02 0A   90
1399 : CD 01 0A D0 05 A9 00 8D   3C
13A1 : 02 0A 4C CD 12 B9 84 0A   8A
13A9 : 29 7F 8D 03 0A A9 00 8D   5F
13B1 : 04 0A 4C CD 12 B9 84 0A   9C
13B9 : 29 7F 8D 04 0A A9 00 8D   8F
13C1 : 03 0A 4C CD 12 AD FB 09   27
13C9 : F0 01 60 A9 00 8D 06 0B   22
13D1 : 20 D6 09 C4 49 45 53 45   CE
13D9 : 52 20 D3 41 54 5A 20 48   82
13E1 : 41 54 20 4B 45 49 4E 20   D6
13E9 : D6 45 52 42 21 0D 00 A9   0D
13F1 : 00 8D FA 09 4C 73 11 20   7C
13F9 : D6 09 D3 49 45 20 42 45   5B
1401 : 46 49 4E 44 45 4E 20 53   F6
1409 : 49 43 48 20 49 4E 20 45   1C
1411 : 49 4E 45 52 20 C7 45 46   FF
1419 : 41 45 4E 47 4E 49 53 2D   50
1421 : 20 20 5A 45 4C 4C 45 2E   29
1429 : 0D C5 53 20 49 53 54 20   B2
1431 : 53 45 48 52 20 55 4E 47   F8
1439 : 45 4D 55 45 54 4C 49 43   76
1441 : 48 20 48 49 45 52 21 0D   5A
1449 : 0D 00 60 20 D6 09 D3 49   0A
1451 : 45 20 42 45 46 49 4E 44   50
1459 : 45 4E 20 53 49 43 48 20   48
1461 : 49 4D 20 C7 45 46 41 45   68
1469 : 4E 47 4E 49 53 4B 45 4C   55
1471 : 4C 45 52 2E 0D C5 53 20   47
1479 : 52 49 45 43 48 54 20 4D   6C
1481 : 4F 44 52 49 47 2E 0D 0D   E4
1489 : 00 60 20 D6 09 D3 49 45   7B
1491 : 20 42 45 46 49 4E 44 45   8F
1499 : 4E 20 53 49 43 48 20 49   7F
14A1 : 4D 20 C7 45 46 41 45 4E   B9
14A9 : 47 4E 49 53 4B 45 4C 4C   7D
14B1 : 45 52 2E 0D D3 49 45 20   29
14B9 : 48 4F 45 52 45 4E 20 D7   3B
14C1 : 41 53 53 45 52 52 41 55   91
14C9 : 53 43 48 45 4E 2E 0D 0D   1D
14D1 : 00 60 20 D6 09 D3 49 45   C3
14D9 : 20 42 45 46 49 4E 44 45   D7
14E1 : 4E 20 53 49 43 48 20 49   C7
14E9 : 4D 20 C7 45 46 41 45 4E   01
14F1 : 47 4E 49 53 4B 45 4C 4C   C5
14F9 : 45 52 2E 0D C9 52 47 45   6B
1501 : 4E 44 57 4F 20 53 43 48   6B
1509 : 52 45 49 54 20 4A 45 4D   DF
1511 : 41 4E 44 2E 0D 0D 00 60   4A
1519 : 20 D6 09 D3 49 45 20 42   25
1521 : 45 46 49 4E 44 45 4E 20   8D
1529 : 53 49 43 48 20 49 4E 20   C1
1531 : 44 45 52 20 45 48 45 4D   F7
1539 : 41 4C 49 47 45 4E 20 20   63
1541 : 20 20 20 C6 4F 4C 54 45   85
1549 : 52 4B 41 4D 4D 45 52 2E   DF
1551 : 0D C5 53 20 53 49 4E 44   5B
1559 : 20 41 42 45 52 20 4B 45   31
1561 : 49 4E 45 20 D7 45 52 4B   AE
1569 : 5A 45 55 47 45 20 4D 45   B9
1571 : 48 52 20 56 4F 52 2D 20   32
1579 : 20 48 41 4E 44 45 4E 21   C1
1581 : 0D 0D 00 60 20 D6 09 D3   A5
1589 : 49 45 20 42 45 46 49 4E   0E
1591 : 44 45 4E 20 53 49 43 48   2D
1599 : 20 49 4E 20 44 45 52 20   ED
15A1 : CC 45 49 43 48 45 4E 48   43
15A9 : 41 4C 4C 45 20 20 20 44   D8
15B1 : 45 53 20 C7 45 46 41 45   B7
15B9 : 4E 47 4E 49 53 53 45 53   F3
15C1 : 2E 0D C5 49 4E 20 C8 41   9C
15C9 : 55 46 45 4E 20 CC 45 49   6D
15D1 : 43 48 45 4E 20 4C 49 45   68
15D9 : 47 45 4E 20 48 45 52 55   FD
15E1 : 4D 2C 20 41 42 45 52 20   4C
15E9 : 20 20 41 4C 4C 45 20 53   09
15F1 : 49 4E 44 20 42 45 52 45   99
15F9 : 49 54 53 20 56 45 52 57   CD
1601 : 45 53 54 2E 0D 0D 00 60   C5
1609 : 20 D6 09 D3 49 45 20 42   15
1611 : 45 46 49 4E 44 45 4E 20   7D
1619 : 53 49 43 48 20 49 4D 20   AC
1621 : CD 55 45 4C 4C 4B 45 4C   41
1629 : 4C 45 52 0D D3 49 45 20   2B
1631 : 48 4F 45 52 45 4E 20 D3   AB
1639 : 54 49 4D 4D 45 4E 20 56   23
1641 : 4F 4D 20 C8 4F 46 2E 0D   52
1649 : 0D 00 60 20 D6 09 D3 49   0A
1651 : 45 20 42 45 46 49 4E 44   50
1659 : 45 4E 20 53 49 43 48 20   48
1661 : 49 4D 20 C7 45 46 41 45   68
1669 : 4E 47 4E 49 53 48 4F 46   59
1671 : 21 0D D5 4D 20 49 48 4E   42
1679 : 20 48 45 52 55 4D 20 49   2C
1681 : 53 54 20 45 49 4E 45 20   0B
1689 : 48 4F 48 45 20 CD 41 55   54
1691 : 45 52 2E 0D 0D 00 60 20   BF
1699 : D6 09 D3 49 45 20 42 45   FB
16A1 : 46 49 4E 44 45 4E 20 53   96
16A9 : 49 43 48 20 49 4D 20 C8   BB
16B1 : 49 4E 52 49 43 48 54 55   52
16B9 : 4E 47 53 52 41 55 4D 2E   1A
16C1 : 0D D6 4F 52 20 49 48 4E   62
16C9 : 45 4E 20 49 53 54 20 45   49
16D1 : 49 4E 20 47 52 4F 53 53   C6
16D9 : 45 52 20 C7 41 4C 47 45   66
16E1 : 4E 21 0D 0D 00 60 20 D6   D6
16E9 : 09 D3 49 45 20 42 45 46   BD
16F1 : 49 4E 44 45 4E 20 53 49   E1
16F9 : 43 48 20 49 4D 20 C6 4C   1B
1701 : 55 52 20 56 4F 52 20 44   E3
1709 : 45 4E 20 20 20 20 20 20   45
1711 : 20 DA 45 4C 4C 45 4E 2E   FE
1719 : 0D D6 49 45 4C 45 20 41   7E
1721 : 42 47 45 53 43 48 4C 4F   09
1729 : 53 53 45 4E 45 20 D4 55   94
1731 : 45 52 45 4E 20 53 54 41   2B
1739 : 52 52 45 4E 20 D3 49 45   20
1741 : 20 41 4E 2E 0D 0D 00 60   55
1749 : 20 D6 09 D3 49 45 20 42   55
1751 : 45 46 49 4E 44 45 4E 20   BD
1759 : 53 49 43 48 20 49 4E 20   F1
```

```
1761 : 45 49 4E 45 52 20 CD 41  67
1769 : 55 45 52 4E 49 53 43 48  8C
1771 : 45 20 20 44 45 52 20 C7  4E
1779 : 45 46 41 45 4E 47 4E 49  C5
1781 : 53 4D 41 55 45 52 21 0D  FB
1789 : CD 41 4E 20 4B 41 4E 4E  23
1791 : 20 53 49 45 20 48 49 45  4A
1799 : 52 20 4E 49 43 48 54 20  C0
17A1 : 53 45 48 45 4E 21 0D 0D  8E
17A9 : 00 60 20 D6 09 D3 49 45  9B
17B1 : 20 42 45 46 49 4E 44 45  AF
17B9 : 4E 20 53 49 43 48 20 49  9F
17C1 : 4E 20 44 45 52 20 C5 49  A9
17C9 : 4E 47 41 4E 47 53 2D 20  D9
17D1 : 20 20 20 20 20 53 43 48  48
17D9 : 4C 45 55 53 45 2E 0D C4  0B
17E1 : 41 53 20 C8 41 55 50 54  96
17E9 : 54 4F 52 20 49 53 54 20  3E
17F1 : 5A 55 21 0D 0D 00 60 20  72
17F9 : D6 09 D3 49 45 20 42 45  5B
1801 : 46 49 4E 44 45 4E 20 53  F6
1809 : 49 43 48 20 41 55 46 20  22
1811 : 44 45 52 20 CD 41 55 45  57
1819 : 52 21 0D C9 4D 20 CF 53  34
1821 : 54 45 4E 20 4C 49 45 47  62
1829 : 54 20 45 49 4E 20 D7 41  D0
1831 : 4C 44 21 0D 0D 00 60 20  1C
1839 : D6 09 D3 49 45 20 42 45  9B
1841 : 46 49 4E 44 45 4E 20 53  36
1849 : 49 43 48 20 49 4E 20 45  5C
1851 : 49 4E 45 4D 20 CE 41 44  C2
1859 : 45 4C 57 41 4C 44 21 0D  48
1861 : C4 45 52 20 C2 4F 44 45  A3
1869 : 4E 20 49 53 54 20 4D 49  92
1871 : 54 20 42 52 41 55 4E 45  33
1879 : 4E 20 CE 41 44 45 4C 4E  EF
1881 : 20 44 49 43 48 54 20 20  66
1889 : 42 45 44 45 43 4B 54 2E  64
1891 : 0D 0D 00 60 20 D6 09 D3  B5
1899 : 49 45 20 42 45 46 49 4E  1E
18A1 : 44 45 4E 20 53 49 43 48  3D
18A9 : 20 41 4E 20 45 49 4E 45  64
18B1 : 52 20 C6 45 4C 53 45 4E  7F
18B9 : 4B 55 45 53 54 45 2E 0D  AD
18C1 : C4 49 45 20 C2 52 41 4E  E0
18C9 : 44 55 4E 47 20 49 53 54  77
18D1 : 20 4D 4F 45 52 44 45 52  15
18D9 : 49 53 43 48 21 0D 0D 00  54
18E1 : 60 4C 95 18 4C 95 18 4C  3A
18E9 : 38 18 4C 38 18 4C 38 18  3C
18F1 : 4C 38 18 4C 38 18 20 D6  5B
18F9 : 09 D3 49 45 20 42 45 46  9D
1901 : 49 4E 44 45 4E 20 53 49  F1
1909 : 43 48 20 41 55 46 20 45  33
1911 : 49 4E 45 52 20 D3 54 52  B4
1919 : 41 53 53 45 2E 0D D3 49  AF
1921 : 45 20 48 4F 45 52 45 4E  0B
1929 : 20 45 49 4E 45 4E 20 C8  E1
1931 : 55 4E 44 20 42 45 4C 4C  DB
1939 : 45 4E 21 0D 0D 00 60 4C  7A
1941 : F7 18 4C 38 18 4C 38 18  53
1949 : 4C F7 18 4C 38 18 4C 38  06
1951 : 18 4C F7 18 4C 38 18 20  B7
1959 : D6 09 D3 49 45 20 42 45  BB
1961 : 46 49 4E 44 45 4E 20 53  56
1969 : 49 43 48 20 42 45 49 20  1E
1971 : 45 49 4E 45 4D 20 D3 54  65
1979 : 45 47 2E 0D C9 4D 20 CE  B4
1981 : 4F 52 44 45 4E 20 53 45  71
1989 : 48 45 4E 20 53 49 45 20  E0
1991 : 45 49 4E 45 20 CE 41 43  BB
1999 : 48 42 41 52 49 4E 53 45  7C
19A1 : 4C 2E 0D 0D 00 60 20 D6  1B
19A9 : 09 D3 49 45 20 42 45 46  4D
19B1 : 49 4E 44 45 4E 20 53 49  A1
19B9 : 43 48 20 41 55 46 20 48  E9
19C1 : 4F 48 45 52 20 D3 45 45  10
19C9 : 2E 0D C9 4D 20 CE 4F 52  F4
19D1 : 44 45 4E 20 49 53 54 20  10
19D9 : 45 49 4E 45 20 47 52 4F  23
19E1 : 53 53 45 20 C9 4E 53 45  1A
19E9 : 4C 21 0D 0D 00 60 4C 95  0A
19F1 : 18 4C 95 18 20 D6 09 D3  1C
19F9 : 49 45 20 42 45 46 49 4E  7E
1A01 : 44 45 4E 20 53 49 43 48  9D
1A09 : 20 41 4E 20 45 49 4E 45  C4
1A11 : 4D 20 4B 4C 45 49 4E 45  2D
1A19 : 4E 20 20 20 20 20 20 C1  8B
1A21 : 4E 4C 45 47 45 52 2E 0D  89
1A29 : C9 4D 20 D3 55 45 44 45  36
1A31 : 4E 20 53 45 48 45 4E 20  35
1A39 : 53 49 45 20 45 49 4E 45  E9
1A41 : 20 4B 4C 45 49 4E 45 20  1F
1A49 : C9 4E 53 45 4C 21 0D 0D  D3
1A51 : 00 60 20 D6 09 D3 49 45  43
1A59 : 20 42 45 46 49 4E 44 45  57
1A61 : 4E 20 53 49 43 48 20 41  37
1A69 : 55 46 20 44 45 4D 20 CD  4D
1A71 : 41 52 4B 54 50 4C 41 54  4E
```

```
1A79 : 5A 0D 45 49 4E 45 53 20  71
1A81 : 4B 4C 45 49 4E 45 4E 20  F5
1A89 : CF 52 54 45 53 2E 0D C5  A5
1A91 : 49 4E 20 D7 45 47 57 45  7B
1A99 : 49 53 45 52 20 5A 55 4D  EC
1AA1 : 20 C6 4C 55 47 50 4C 41  8D
1AA9 : 54 5A 20 5A 45 49 47 54  E2
1AB1 : 20 4E 41 43 48 20 20 CF  57
1AB9 : 53 54 45 4E 21 0D 0D 00  00
1AC1 : 60 20 D6 09 D3 49 45 20  E5
1AC9 : 42 45 46 49 4E 44 45 4E  21
1AD1 : 20 53 49 43 48 20 49 4D  9B
1AD9 : 20 D2 45 53 54 41 55 52  67
1AE1 : 41 4E 54 2E 0D C5 53 20  B1
1AE9 : 44 55 46 54 45 54 20 47  FA
1AF1 : 55 54 20 4E 41 43 48 20  D2
1AF9 : C5 53 53 45 4E 21 0D 0D  22
1B01 : 00 60 20 D6 09 D3 49 45  F3
1B09 : 20 42 45 46 49 4E 44 45  07
1B11 : 4E 20 53 49 43 48 20 41  E7
1B19 : 55 46 20 44 45 52 20 D3  31
1B21 : 54 52 41 53 53 45 2E 0D  BB
1B29 : C5 53 20 47 49 42 54 20  C1
1B31 : 48 49 45 52 20 4B 45 49  BD
1B39 : 4E 45 20 C1 55 54 4F 53  46
1B41 : 2E 0D 0D 00 60 20 D6 09  AE
1B49 : D3 49 45 20 53 54 45 48  94
1B51 : 45 4E 20 49 4E 20 44 45  70
1B59 : 52 20 CB 55 45 43 48 45  73
1B61 : 20 45 49 4E 45 53 20 D0  51
1B69 : 52 49 56 41 54 2D 20 20  8D
1B71 : 48 41 55 53 45 53 2E 0D  DB
1B79 : D6 4F 52 20 49 48 4E 45  2A
1B81 : 4E 20 53 54 45 48 54 20  67
1B89 : 45 49 4E 20 53 43 48 57  2A
1B91 : 45 52 45 52 20 D4 49 53  0F
1B99 : 43 48 2E 0D 0D 00 60 20  C0
1BA1 : D6 09 D3 49 45 20 42 45  03
1BA9 : 46 49 4E 44 45 4E 20 53  9E
1BB1 : 49 43 48 20 41 55 46 20  CA
1BB9 : 45 49 4E 45 52 20 CC 41  BB
1BC1 : 4E 44 53 54 52 41 53 53  B4
1BC9 : 45 2E C1 4E 20 44 45 4E  35
1BD1 : 20 D3 45 49 54 45 4E 20  3E
1BD9 : 53 54 45 48 45 4E 20 C2  7E
1BE1 : 41 45 55 4D 45 2E 0D 0D  D8
1BE9 : 00 60 20 D6 09 D3 49 45  DB
1BF1 : 20 53 54 45 48 45 4E 20  A1
1BF9 : 49 4D 20 D6 4F 52 47 41  F3
1C01 : 52 54 45 4E 2E 20 45 49  86
1C09 : 45 53 20 C5 43 48 47 52  09
1C11 : 55 4E 44 2D 20 20 53 54  DD
1C19 : 45 43 4B 45 53 2E 0D C4  E0
1C21 : 41 53 20 C8 41 55 53 20  79
1C29 : 49 53 54 20 56 45 52 53  B4
1C31 : 43 48 4C 4F 53 53 45 4E  17
1C39 : 2E 0D 0D 00 60 4C 03 1B  DC
1C41 : 20 D6 09 D3 49 45 20 42  4D
1C49 : 45 46 49 4E 44 45 4E 20  B5
1C51 : 53 49 43 48 20 49 4E 20  E9
1C59 : 45 49 4E 45 52 20 53 43  79
1C61 : 48 4D 41 4C 45 4E 20 20  B1
1C69 : 20 20 20 C7 41 53 53 45  21
1C71 : 2E 0D 0D 00 60 20 D6 09  DE
1C79 : D3 49 45 20 53 49 4E 44  87
1C81 : 20 49 4E 20 45 49 4E 45  40
1C89 : 52 20 C1 50 4F 54 48 45  A9
1C91 : 4B 45 2E 0D D5 45 42 45  C7
1C99 : 52 41 4C 4C 20 53 54 45  A1
1CA1 : 48 45 4E 20 CD 45 44 49  CE
1CA9 : 4B 41 4D 45 4E 54 45 2E  8A
1CB1 : 0D 0D 00 60 20 D6 09 D3  D5
1CB9 : 49 45 20 53 49 4E 44 20  70
1CC1 : 49 4E 20 45 49 4E 45 52  A3
1CC9 : 20 CE 41 43 48 54 42 41  BC
1CD1 : 52 2E 0D C4 45 52 20 D2  23
1CD9 : 41 55 53 53 43 48 4D 45  3A
1CE1 : 49 53 53 45 52 20 57 41  57
1CE9 : 4C 54 45 54 20 53 45 49  7F
1CF1 : 4E 45 53 20 C1 4D 54 45  1D
1CF9 : 53 2E 0D 0D 00 60 20 D6  7A
1D01 : 09 D3 49 45 20 53 49 4E  4D
1D09 : 44 20 49 4E 20 45 49 4E  67
1D11 : 45 4D 20 CB 4C 45 49 44  1B
1D19 : 45 52 47 45 53 43 48 41  F5
1D21 : 45 46 54 21 0D D6 4F 52  2C
1D29 : 20 49 48 4E 45 4E 20 53  B8
1D31 : 54 45 48 54 20 45 49 4E  B2
1D39 : 20 D3 43 48 49 4C 44 3A  99
1D41 : 0D CE C5 C8 CD C5 20 CE  69
1D49 : D5 D2 20 D7 C5 D2 D4 D3  78
1D51 : C1 C3 C8 C5 CE 20 C1 CC  6D
1D59 : D3 20 DA C1 C8 CC D5 CE  13
1D61 : C7 D3 CD C9 D4 D4 C5 CC  63
1D69 : 0D 0D 00 60 4C 03 1B 20  93
1D71 : D6 09 D3 49 45 20 53 49  1F
1D79 : 4E 44 20 49 4E 20 45 49  A8
1D81 : 4E 45 52 20 C7 4F 4C 44  BB
1D89 : 53 43 48 4D 49 45 44 45  94
```

```
1D91 : 21 0D CE 49 45 4D 41 4E  76
1D99 : 44 20 49 53 54 20 5A 55  04
1DA1 : 20 53 45 48 45 4E 21 0D  2B
1DA9 : 0D 00 60 20 D6 09 D3 49  6A
1DB1 : 45 20 42 45 46 49 4E 44  B0
1DB9 : 45 4E 20 53 49 43 48 20  AB
1DC1 : 49 4E 20 45 49 4E 45 52  A3
1DC9 : 20 D6 4F 52 4F 52 54 2D  A6
1DD1 : 20 20 20 20 20 53 49 45  F0
1DD9 : 45 44 4C 55 4E 47 2E 0D  F0
1DE1 : C9 4D 20 CE 4F 52 44 45  56
1DE9 : 4E 20 52 41 55 53 43 48  92
1DF1 : 54 20 44 41 53 20 CD 45  87
1DF9 : 45 52 21 0D 0D 00 60 20  E4
1E01 : D6 09 D3 49 45 20 53 49  AF
1E09 : 4E 44 20 41 4D 20 C2 41  0D
1E11 : 44 45 53 54 41 4E 44 2E  4B
1E19 : 0D C8 49 45 52 20 49 53  77
1E21 : 54 20 53 43 48 4F 45 4E  73
1E29 : 45 52 20 57 45 49 43 48  C7
1E31 : 45 52 20 D3 41 4E 44 2E  16
1E39 : 0D 0D 00 60 20 D6 09 D3  5D
1E41 : 49 45 20 53 49 4E 44 20  F8
1E49 : 41 4D 20 D3 54 52 41 4E  2D
1E51 : 44 21 0D C5 53 20 49 53  24
1E59 : 54 20 45 49 4E 53 41 4D  57
1E61 : 20 48 49 45 52 21 0D 0D  1D
1E69 : 00 60 4C A0 1B 4C A0 1B  8D
1E71 : 4C A0 1B 4C A0 1B 4C A0  B3
1E79 : 1B 4C A0 1B 20 D6 09 D3  CA
1E81 : 49 45 20 53 49 4E 44 20  38
1E89 : 49 4E 20 45 49 4E 45 52  6B
1E91 : 20 D3 43 48 45 55 4E 45  37
1E99 : 2E 0D C8 49 45 52 20 49  A3
1EA1 : 53 54 20 4E 55 52 20 D3  00
1EA9 : 54 52 4F 48 2E 0D 0D 00  83
1EB1 : 60 20 D6 09 D3 49 45 20  D5
1EB9 : 53 49 4E 44 20 41 55 46  BB
1EC1 : 20 45 49 4E 45 4D 20 C2  65
1EC9 : 41 55 45 52 4E 48 4F 46  41
1ED1 : 2E 0D D3 49 45 20 52 49  D5
1ED9 : 45 43 48 45 4E 20 47 55  28
1EE1 : 54 45 20 CC 41 4E 44 4C  AA
1EE9 : 55 46 54 2E 0D 0D 00 60  36
1EF1 : 4C A0 1B 20 D6 09 D3 49  F0
1EF9 : 45 20 53 49 4E 44 20 41  56
1F01 : 55 46 20 45 49 4E 45 52  EB
1F09 : 20 CB 52 45 55 5A 55 4E  66
1F11 : 47 2E 0D C5 49 4E 20 D7  A3
1F19 : 45 47 57 45 49 53 45 52  69
1F21 : 20 5A 55 4D 20 C6 4C 55  81
1F29 : 47 50 4C 41 54 5A 20 57  1B
1F31 : 45 49 53 54 20 4E 41 43  7A
1F39 : 48 20 20 CF 53 54 45 4E  1D
1F41 : 21 0D 0D 00 60 20 D6 09  A1
1F49 : D3 49 45 20 53 49 4E 44  57
1F51 : 20 41 4E 20 45 49 4E 45  0C
1F59 : 52 20 D3 54 45 49 4C 4B  A1
1F61 : 55 45 53 54 45 2E 0D 0D  CC
1F69 : 00 60 20 D6 09 D3 49 45  5B
1F71 : 20 53 54 45 48 45 4E 20  21
1F79 : 41 55 46 20 45 49 4E 45  5D
1F81 : 4D 20 53 43 48 4D 41 4C  AB
1F89 : 45 4E 20 C6 55 53 53 57  C2
1F91 : 45 47 2E 0D C5 52 20 49  A9
1F99 : 53 54 20 4E 49 43 48 54  61
1FA1 : 20 47 45 50 46 4C 41 53  33
1FA9 : 54 45 52 54 2E 0D 0D 00  3E
1FB1 : 60 4C 6B 1F 20 D6 09 D3  7B
1FB9 : 49 45 20 53 49 4E 44 20  70
1FC1 : 49 4D 20 D7 41 53 53 45  3A
1FC9 : 52 57 45 52 4B 2E 0D C8  4E
1FD1 : 49 45 52 20 53 54 45 48  D3
1FD9 : 45 4E 20 56 49 45 4C 45  93
1FE1 : 20 47 52 4F 53 53 45 20  48
1FE9 : D4 41 4E 4B 53 2E 0D 0D  50
1FF1 : 00 60 4C A0 1B 4C A0 1B  15
1FF9 : 20 D6 09 D3 49 45 20 53  27
2001 : 49 4E 44 20 49 4E 20 45  99
2009 : 49 4E 45 4D 20 CC 41 55  8C
2011 : 42 57 41 4C 44 2E 0D C5  4E
2019 : 53 20 49 53 54 20 56 45  63
2021 : 52 44 41 45 43 48 54 49  E9
2029 : 47 20 53 54 49 4C 4C 21  4A
2031 : 0D 0D 00 60 4C F9 1F 4C  7B
2039 : F9 1F 4C F9 1F 20 D6 09  74
2041 : D3 49 45 20 53 49 4E 44  4F
2049 : 20 49 4E 20 45 49 4E 45  08
2051 : 52 20 D0 4F 4C 49 5A 45  D4
2059 : 49 4B 4F 4E 54 52 4F 4C  93
2061 : 4C 45 21 21 21 0D C7 4C  EF
2069 : 45 49 43 48 20 57 49 52  B3
2071 : 44 20 4D 41 4E 20 53 49  07
2079 : 43 48 20 41 4E 20 53 49  D6
2081 : 45 20 57 45 4E 44 45 4E  0E
2089 : 21 0D 0D 00 60 4C F9 1F  03
2091 : 4C F9 1F 4C A0 1B 4C F9  33
2099 : 1F 4C F9 1F 20 D6 09 D3  C5
20A1 : 49 45 20 53 49 4E 44 20  58
```

## Listing »Freiheit« (Fortsetzung)

```
20A9 : 41 55 46 20 45 49 4E 45  8D
20B1 : 52 20 C2 52 55 45 43 4B  31
20B9 : 45 2E 0D C4 41 52 55 4E  8A
20C1 : 54 45 52 20 46 4C 49 45  C7
20C9 : 53 53 54 20 45 49 4E 20  F7
20D1 : 42 52 45 49 54 45 52 20  B0
20D9 : C6 4C 55 53 53 2E 0D 0D  7A
20E1 : 00 60 20 D6 09 D3 49 45  D3
20E9 : 20 53 49 4E 44 20 41 4E  B6
20F1 : 20 45 49 4E 45 4D 20 C2  95
20F9 : 52 45 49 54 45 4E 20 C6  A0
2101 : 4C 55 53 53 2C 20 4D 49  C3
2109 : 54 20 20 20 20 56 49 45  DE
2111 : 4C 45 4E 20 D3 54 52 4F  5F
2119 : 4D 53 43 48 4E 45 4C 4C  C3
2121 : 45 4E 21 0D 0D 00 60 4C  62
2129 : F9 1F 4C E3 20 20 D6 09  B2
2131 : D3 49 45 20 53 49 4E 44  3F
2139 : 20 49 4E 20 45 49 4E 45  F8
2141 : 4D 20 47 52 4F 53 53 45  22
2149 : 4E 20 D3 55 4D 50 46 21  FA
2151 : 0D CD 41 4E 20 56 45 52  CD
2159 : 53 49 4E 4B 54 20 4B 4E  5E
2161 : 49 45 54 49 45 46 21 0D  B0
2169 : 0D 00 60 4C A0 1B 4C 2E  88
2171 : 21 20 D6 09 D3 49 45 20  56
2179 : 53 49 4E 44 20 49 4E 20  53
2181 : 45 49 4E 45 52 20 4B 4C  93
2189 : 45 49 4E 45 4E 20 CE 49  63
2191 : 45 53 43 48 45 2E 0D 0D  6E
2199 : 00 60 20 D6 09 D3 49 45  8B
21A1 : 20 53 49 4E 44 20 49 4E  8E
21A9 : 20 45 49 4E 45 4D 20 4B  5E
21B1 : 55 52 5A 45 4E 20 D4 55  52
21B9 : 4E 4E 45 4C 2E 0D 0D 00  89
21C1 : 60 4C A0 1B 20 D6 09 D3  57
21C9 : 49 45 20 53 49 4E 44 20  80
21D1 : 49 4E 20 45 49 4E 45 52  B3
21D9 : 20 C7 52 4F 54 54 45 2E  B5
21E1 : 0D D4 52 4F 50 46 45 4E  C0
21E9 : 20 46 41 4C 4C 45 4E 20  6F
21F1 : 41 55 46 20 44 45 4E 20  5A
21F9 : C2 4F 44 45 4E 2E 0D 0D  C1
2201 : 00 60 20 D6 09 D3 49 45  F3
2209 : 20 42 45 46 49 4E 44 45  07
2211 : 4E 20 53 49 43 48 20 49  F7
2219 : 4E 20 45 49 4E 45 4D 20  76
2221 : C6 45 4C 53 53 50 41 4C  5D
2229 : 54 2E 0D 0D 00 60 20 D6  AB
2231 : 09 D3 49 45 20 42 45 46  D5
2239 : 49 4E 44 45 4E 20 53 49  29
2241 : 43 48 20 49 4E 20 45 49  67
2249 : 4E 45 52 20 47 52 4F 53  BD
2251 : 53 45 4E 20 20 20 20 20  A2
2259 : 20 C5 42 45 4E 45 2E 0D  77
2261 : C9 4D 20 D3 55 45 44 45  6E
2269 : 4E 20 48 4F 45 52 45 4E  5C
2271 : 20 53 49 45 20 C6 4C 55  4A
2279 : 47 4C 41 45 52 4D 21 0D  0E
2281 : 0D 00 60 20 D6 09 D3 49  42
2289 : 45 20 53 54 45 48 45 4E  86
2291 : 20 41 55 46 20 44 45 4D  44
2299 : 20 C6 4C 55 47 46 45 4C  2E
22A1 : 44 2E 0D C4 49 52 45 4B  AB
22A9 : 54 20 4E 45 42 45 4E 20  11
22B1 : 49 48 4E 45 4E 20 4C 41  F4
22B9 : 4E 44 45 54 20 45 49 4E  F3
22C1 : 20 CA 55 4D 42 4F 2E 0D  B7
22C9 : 0D 00 60 20 D6 09 D3 49  8A
22D1 : 45 20 53 49 4E 44 20 49  3E
22D9 : 4E 20 44 45 52 20 C6 4C  CB
22E1 : 55 47 48 41 46 45 4E 48  6C
22E9 : 41 4C 4C 45 21 0D C8 49  3C
22F1 : 45 52 20 48 45 52 53 43  2B
22F9 : 48 54 20 C8 4F 43 48 42  41
2301 : 45 54 52 49 45 42 2E 0D  67
2309 : 0D 00 60 20 D6 09 D3 49  CA
2311 : 45 20 53 49 54 5A 45 4E  2E
2319 : 20 49 4D 20 C6 4C 55 47  E8
2321 : 5A 45 55 47 21 0D C4 45  74
2329 : 52 20 D3 49 54 5A 20 49  D5
2331 : 53 54 20 53 45 48 52 20  41
2339 : 42 45 51 55. 45 4D 21 0D  7A
2341 : 0D 00 60 20 D6 09 D3 49  02
2349 : 45 20 53 54 45 48 45 4E  46
2351 : 20 41 4D 20 D3 43 48 41  64
2359 : 4C 54 45 52 21 0D CD 41  9F
2361 : 4E 20 4B 41 4E 4E 20 48  23
2369 : 49 45 52 20 C6 4C 55 47  A0
2371 : 54 49 43 4B 45 54 53 20  29
2379 : 4B 41 55 46 45 4E 21 0D  E8
2381 : 0D 00 60 20 D6 09 D3 49  42
2389 : 45 20 42 45 46 49 4E 44  88
2391 : 45 4E 20 53 49 43 48 20  80
2399 : 49 4E 20 45 49 4E 45 4D  71
23A1 : 20 CD 4F 54 4F 52 42 4F  35
23A9 : 4F 54 21 0D C5 53 20 53  2A
23B1 : 43 48 41 55 4B 45 4C 54  CC
23B9 : 20 53 41 4E 46 54 20 41  A7
23C1 : 55 46 20 44 45 4E 20 D7  C1
23C9 : 45 4C 4C 45 4E 2E 0D 0D  95
23D1 : 00 60 01 00 00 0B 34 14  93
23D9 : 0C 2D 0A 01 32 09 03 04  9E
23E1 : 09 31 0E 27 3B 26 3E 54  72
23E9 : 5B 02 00 03 00 03 03 00  CA
23F1 : 03 01 00 03 01 01 00 00  ED
23F9 : 00 00 00 00 00 00 00 01  FC
2401 : 00 42 08 20 07 45 2A 40  EC
2409 : 24 31 33 2C 20 01 00 00  22
2411 : 00 00 00 00 00 00 00 00  12
2419 : 00 00 00 00 00 00 00 00  1A
2421 : 00 00 00 00 00 00 00 00  22
2429 : 20 00 00 00 01 05 03 0A  A2
2431 : 0C 0F 22 26 2F 28 3C 42  BC
2439 : 49 49 29 44 46 5E 5F 5C  87
2441 : 5D 5E 25 00 00 00 00 00  17
2449 : 00 00 00 00 00 00 00 00  4A
2451 : 00 00 00 00 00 00 00 00  52
2459 : 00 00 00 00 00 00 00 00  5A
2461 : 00 00 00 00 00 00 00 00  62
2469 : 00 00 00 00 00 00 00 00  6A
2471 : 00 00 00 00 00 00 40 00  73
2479 : 00 00 00 00 00 00 00 00  7A
2481 : 00 00 00 00 00 00 00 00  82
2489 : 00 00 00 00 00 00 EA 20  75
2491 : 00 78 AE D3 23 BD A2 24  E6
2499 : 85 14 BD 02 25 85 15 6C  84
24A1 : 14 00 F8 4C 8B D3 19 85  44
24A9 : 09 4C 98 E7 49 AB F8 38  42
24B1 : 95 E2 E5 E8 EB EE F1 F4  36
24B9 : F7 40 43 46 49 4C 4F 52  43
24C1 : 55 58 A7 EF F2 F5 53 C2  DC
24C9 : 03 46 A0 EB 3E 41 76 B5  C8
24D1 : FF 6D 70 AC 00 3D 6B 6E  AD
24D9 : 71 74 77 7A 7D B2 F1 F4  D1
24E1 : 46 6B B2 B5 F3 F6 F9 35  89
24E9 : 38 3B 3E 8E 91 94 97 9A  72
24F1 : 9D E3 28 2B 2E 6C 6F 72  D8
24F9 : 9B C2 C5 03 2F 84 CC 0C  29
2501 : 44 84 13 14 14 14 15 15  2F
2509 : 16 16 16 16 17 17 17 18  29
2511 : 18 18 18 18 18 18 18 18  11
2519 : 18 19 19 19 19 19 19 19  18
2521 : 19 19 19 19 19 19 1A 1A  27
2529 : 1B 1B 1B 1B 1C 1C 1C 1C  47
2531 : 1C [illegible] 1D 1D 1E 1E 1E 1E  4E
2539 : 1E 1E 1E 1E 1E 1E 1E 1E  39
2541 : 1F 1F 1F 1F 1F 1F 1F 20  43
2549 : 20 20 20 20 20 20 20 20  49
2551 : 20 20 21 21 21 21 21 21  D0
2559 : 21 21 21 22 22 22 22 23  99
2561 : 23 23 00 05 07 06 00 00  57
2569 : 00 0C 00 00 00 00 00 16  9C
2571 : 0E 12 13 15 14 00 19 1A  CA
2579 : 1B 1C 1F 1E 1D 23 00 00  19
2581 : 20 21 24 00 00 25 27 00  01
2589 : 2B 00 00 2C 30 2F 00 00  B6
2591 : 00 32 00 33 00 00 00 00  11
2599 : 00 00 00 3B 00 00 3C 39  64
25A1 : 3E 00 00 00 48 4A 00 00  B6
25A9 : 46 47 4B 00 00 4D 4E 00  09
25B1 : 00 51 00 00 00 53 54 00  46
25B9 : 56 57 00 00 58 5B 00 00  1B
25C1 : 5D 00 00 00 00 02 04 03  45
25C9 : 00 00 00 00 08 00 0F 00  86
25D1 : 00 00 10 11 13 12 0E 00  F2
25D9 : 00 15 16 17 18 1B 1A 19  C1
25E1 : 1F 20 00 1C 21 24 00 25  11
25E9 : 00 00 00 27 2A 00 00 2C  C9
25F1 : 2B 00 30 32 00 00 00 00  6F
25F9 : 00 3E 00 3A 3D 00 3F 00  31
2601 : 00 00 00 00 00 00 47 48  AF
2609 : 43 00 44 49 00 4C 4D 00  1E
2611 : 00 50 00 54 55 00 57 58  27
2619 : 5B 00 00 5C 00 5F 00 00  FB
2621 : 00 04 02 00 06 00 05 0A  2D
2629 : 00 00 08 00 0B 00 00 0F  FA
2631 : 10 0E 12 15 16 17 18 0C  02
2639 : 1A 1B 1C 22 23 1D 1E 00  C0
2641 : 00 00 00 00 26 00 28 00  45
2649 : 25 27 2D 2B 00 2C 00 31  76
2651 : 00 00 00 33 00 35 00 00  62
2659 : 00 29 00 3B 3A 3D 40 41  66
2661 : 42 00 3E 48 38 37 45 39  01
2669 : 44 47 4A 49 4B 00 4C 4D  8D
2671 : 4E 51 00 4F 00 00 54 00  A3
2679 : 57 58 00 00 00 00 00 00  FD
2681 : 03 00 02 07 05 00 0B 00  62
2689 : 08 0D 00 0E 12 10 11 00  C0
2691 : 13 00 00 14 15 16 17 00  85
2699 : 19 1A 1B 1E 1F 00 00 00  3C
26A1 : 1C 1D 00 29 25 2A 27 3A  26
26A9 : 00 2C 2E 2B 00 00 00 30  11
26B1 : 00 34 00 36 00 46 45 48  6A
26B9 : 3D 3C 00 3E 43 00 3F 40  8E
26C1 : 41 00 49 47 00 4A 44 4C  3A
26C9 : 4B 4D 4F 50 51 54 00 52  F5
26D1 : 00 00 57 00 00 59 5A 00  DB
26D9 : 00 00 00 00 00 00 00 00  DA
26E1 : 00 00 00 00 00 00 00 00  E2
26E9 : 00 00 00 00 00 0E 12 16  CE
26F1 : 15 00 1A 00 00 00 00 00  8D
26F9 : 00 00 00 00 00 00 00 00  FA
2701 : 00 00 00 00 00 00 00 00  02
2709 : 00 00 00 00 00 00 00 00  0A
2711 : 00 00 34 00 36 00 38 00  63
2719 : 00 00 00 00 00 00 00 25  64
2721 : 00 00 00 00 00 45 00 46  D8
2729 : 00 00 00 00 00 00 00 00  2A
2731 : 00 00 00 00 00 00 00 00  32
2739 : 00 00 5B 00 00 00 00 00  10
2741 : 00 00 00 00 00 00 00 00  42
2749 : 00 00 15 12 13 00 14 00  53
2751 : 00 00 19 00 00 00 1F 1E  50
2759 : 1D 00 20 00 00 00 23 00  0B
2761 : 00 00 00 00 00 00 00 00  62
2769 : 00 00 00 00 00 00 34 00  3A
2771 : 00 00 00 00 37 00 00 00  E5
2779 : 00 00 00 00 00 00 00 44  02
2781 : 00 46 00 00 4A 00 00 4E  E6
2789 : 00 00 00 00 52 00 00 00  AF
2791 : 00 00 00 00 00 00 00 00  92
2799 : 00 00 00 00 00 00 00 00  9A
27A1 : 00 00 00 00 00 00 00 00  A2
27A9 : 00 00 00 00 00 0F 10 12  86
27B1 : 0E 00 00 00 16 00 00 00  21
27B9 : 1C 1B 1A 1E 00 00 22 00  36
27C1 : 00 00 00 00 00 00 00 00  C2
27C9 : 00 00 00 00 00 00 00 32  2E
27D1 : 00 00 38 00 00 00 00 00  E0
27D9 : 00 00 00 00 00 00 00 43  60
27E1 : 00 45 00 00 00 48 00 00  C6
27E9 : 00 4B 00 00 00 50 00 00  12
27F1 : 00 00 00 00 00 00 00 00  F2
27F9 : 00 00 00 00 00 00 00 00  FA
2801 : 00 00 00 00 00 00 00 00  02
2809 : 10 00 00 00 11 00 00 13  51
2811 : 12 00 00 00 15 00 00 00  75
2819 : 00 00 00 00 00 00 00 42  9E
2821 : 00 00 00 00 00 00 00 00  22
2829 : 00 00 00 00 00 00 35 00  FE
2831 : 37 00 39 00 00 00 00 00  B7
2839 : 00 00 00 00 00 00 00 48  CA
2841 : 4A 00 00 00 00 00 00 00  8C
2849 : 00 00 00 00 00 00 00 00  4A
2851 : 00 00 00 00 00 5D 00 00  3C
2859 : 00 00 00 01 00 00 00 09  8C
2861 : 08 00 00 00 00 00 00 00  6A
2869 : 00 00 00 00 00 00 00 00  6A
2871 : 00 00 00 00 00 00 00 00  72
2879 : 00 00 00 00 00 00 00 00  7A
2881 : 00 00 00 00 00 00 00 00  82
2889 : 00 00 00 00 00 00 00 00  8A
2891 : 00 00 00 00 00 00 00 00  92
2899 : 00 00 00 00 00 00 00 00  9A
28A1 : 00 00 00 00 00 00 00 00  A2
28A9 : 00 00 00 00 00 00 00 00  AA
28B1 : 00 00 00 00 00 00 5E 00  2B
28B9 : 00 02 00 00 00 00 00 00  BB
28C1 : 07 06 00 00 00 00 00 00  CC
28C9 : 00 00 00 00 00 00 00 00  CA
28D1 : 00 00 00 00 00 00 00 00  D2
28D9 : 00 00 00 00 00 00 00 00  DA
28E1 : 00 00 00 00 00 00 00 00  E2
28E9 : 00 00 00 00 00 00 00 00  EA
28F1 : 00 00 00 00 00 00 00 00  F2
28F9 : 00 00 00 00 00 00 00 00  FA
2901 : 00 00 00 00 00 00 00 00  02
2909 : 00 00 00 00 00 00 00 00  0A
2911 : 00 00 00 00 00 00 5D 00  87
2919 : 20 00 00 00 00 00 00 00  3A
2921 : 00 00 00 AE D3 23 BD 62  09
2929 : 25 8D 1A 29 BD C1 25 8D  5A
2931 : 1B 29 BD 20 26 8D 1C 29  E6
2939 : BD 7F 26 8D 1D 29 BD DE  C1
2941 : 26 8D 1E 29 BD 3D 27 8D  58
2949 : 1F 29 BD 9C 27 8D 20 29  B1
2951 : BD FB 27 8D 21 29 BD 5A  8E
2959 : 28 8D 22 29 BD B9 28 8D  5B
2961 : 23 29 60 A9 08 20 D2 FF  33
2969 : A9 0E 20 D2 FF A9 93 20  58
2971 : D2 FF A9 00 8D FA 09 A9  D5
2979 : 00 8D FC 09 8D FD 09 8D  A8
2981 : FF 09 8D FE 09 8D FB 09  47
2989 : A9 06 8D 20 D0 8D 21 D0  3C
2991 : A9 9E 20 D2 FF A2 BB BD  6B
2999 : 7F 2B 9D D2 23 CA D0 F7  2B
29A1 : 20 A8 52 A9 00 8D 12 35  FE
29A9 : 4C 84 6B 20 7C 6B A9 01  E2
29B1 : 8D 12 35 20 73 11 A9 0D  19
29B9 : 20 D2 FF 20 CF 4F EA EA  3F
29C1 : EA EA EA EA EA EA EA EA  C0
29C9 : EA EA EA EA EA EA EA EA  C8
29D1 : EA EA EA EA EA EA EA EA  D0
29D9 : EA EA EA EA EA EA EA EA  D8
```

```
29E1 : EA EA EA EA EA EA EA 20  4A
29E9 : 00 41 20 D7 3D AD 12 35  81
29F1 : F0 B9 20 D6 09 C4 41 53  03
29F9 : 20 4B 4F 45 4E 4E 45 4E  44
2A01 : 20 53 49 45 20 4E 49 43  E6
2A09 : 48 54 21 0D D0 52 55 45  E5
2A11 : 46 45 4E 20 53 49 45 20  66
2A19 : 49 48 52 45 20 C5 49 4E  B6
2A21 : 47 41 42 45 21 21 21 0D  FC
2A29 : 0D 00 A9 00 8D FA 09 4C  0E
2A31 : AF 29 20 D6 09 C9 4E 20  B0
2A39 : 44 49 45 53 45 52 20 D2  EB
2A41 : 49 43 48 54 55 4E 47 20  EE
2A49 : 49 53 54 20 4B 45 49 4E  F6
2A51 : 20 D7 45 47 21 0D 0D 00  46
2A59 : A9 00 8D FA 09 8D 12 35  75
2A61 : 4C 30 77 20 D6 09 CD 4F  33
2A69 : 45 47 4C 49 43 48 45 20  5A
2A71 : C1 55 53 47 41 45 4E 47  A1
2A79 : 45 20 42 5A 57 2E 20 D2  B7
2A81 : 49 43 48 54 55 4E 47 45  78
2A89 : 4E 3A 00 A9 00 8D 09 0B  D0
2A91 : AD 1A 29 F0 0F EE 09 0B  56
2A99 : 20 D6 09 0D CE 4F 52 44  42
2AA1 : 45 4E 2C 00 AD 1B 29 F0  53
2AA9 : 0F EE 09 0B 20 D6 09 0D  CA
2AB1 : D3 55 45 44 45 4E 2C 00  80
2AB9 : AD 1C 29 F0 0F EE 09 0B  7F
2AC1 : 20 D6 09 0D D7 45 53 54  CE
2AC9 : 45 4E 2C 00 AD 1D 29 F0  8B
2AD1 : 0E EE 09 0B 20 D6 09 0D  F1
2AD9 : CF 53 54 45 4E 2C 00 AD  B1
2AE1 : 1E 29 F0 13 EE 09 0B 20  D6
2AE9 : D6 09 0D CE 4F 52 44 57  A8
2AF1 : 45 53 54 45 4E 2C 00 AD  3F
2AF9 : 1F 29 F0 12 EE 09 0B 20  CF
2B01 : D6 09 0D CE 4F 52 44 4F  B0
2B09 : 53 54 45 4E 2C 00 AD 20  5B
2B11 : 29 F0 13 EE 09 0B 20 D6  6C
2B19 : 09 0D D3 55 45 44 57 45  A7
2B21 : 53 54 45 4E 2C 00 AD 21  75
2B29 : 29 F0 12 EE 09 0B 20 D6  44
2B31 : 09 0D D3 55 45 44 4F 53  BB
2B39 : 54 45 4E 2C 00 AD 22 29  91
2B41 : F0 0D EE 09 0B 20 D6 09  B4
2B49 : 0D C8 4F 43 48 2C 00 AD  38
2B51 : 23 29 F0 0F EE 09 0B 20  CA
2B59 : D6 09 0D D2 55 4E 54 45  F5
2B61 : 52 2C 00 AD 09 0B F0 09  3E
2B69 : 20 D6 09 9D 2E 0D 0D 00  6A
2B71 : 60 20 D6 09 0D CB 45 49  8F
2B79 : 4E 45 21 0D 0D 00 60 01  A8
2B81 : 00 00 0B 34 14 0C 2D 0A  35
2B89 : 01 32 09 03 04 09 31 0E  B0
2B91 : 27 3B 26 3E 54 5B 02 00  CF
2B99 : 03 00 03 03 00 03 01 00  DA
2BA1 : 03 01 01 00 00 00 00 00  65
2BA9 : 00 00 00 00 01 00 42 08  D3
2BB1 : 20 07 45 2A 40 24 31 33  3C
2BB9 : 2C 20 01 00 00 00 00 00  36
2BC1 : 00 00 00 00 00 00 00 00  C2
2BC9 : 00 00 00 00 00 00 00 00  CA
2BD1 : 00 00 00 00 00 20 00 00  D3
2BD9 : 00 01 05 03 0A 0C 0F 22  7D
2BE1 : 26 2F 28 3C 42 49 49 29  16
2BE9 : 44 46 5E 5F 5C 5D 5E 25  48
2BF1 : 00 00 00 00 00 00 00 00  F2
2BF9 : 00 00 00 00 00 00 00 00  FA
2C01 : 00 00 00 00 00 00 00 00  02
2C09 : 00 00 00 00 00 00 00 00  0A
2C11 : 00 00 00 00 00 00 00 00  12
2C19 : 00 00 00 00 00 00 00 00  1A
2C21 : 00 00 00 40 00 00 00 00  2A
2C29 : 00 00 00 00 00 00 00 00  2A
2C31 : 00 00 00 00 00 00 00 00  32
2C39 : 00 00 00 00 00 00 00 00  3A
2C41 : 20 D6 09 D3 49 45 20 53  6F
2C49 : 45 48 45 4E 20 C5 44 45  99
2C51 : 2C 20 49 48 52 45 4E 20  B2
2C59 : CD 49 54 47 45 46 41 4E  F1
2C61 : 47 45 4E 45 4E 2E 0D C5  9D
2C69 : 52 20 48 41 54 20 4E 49  18
2C71 : 43 48 54 53 20 42 45 49  14
2C79 : 20 53 49 43 48 21 0D 00  BF
2C81 : 60 20 D6 09 D3 49 45 20  A5
2C89 : 53 45 48 45 4E 20 45 49  C7
2C91 : 4E 45 4E 20 D7 41 45 52  5B
2C99 : 54 45 52 21 0D 00 60 4C  34
2CA1 : 82 2C 4C 82 2C 4C 82 2C  24
2CA9 : 4C 82 2C 4C 82 2C 20 D6  83
2CB1 : 09 D3 49 45 20 53 45 48  E1
2CB9 : 45 4E 20 45 49 4E 45 4E  8F
2CC1 : 20 CB 45 4C 4C 4E 45 52  93
2CC9 : 2E 0D 00 AD DE 23 C9 09  74
2CD1 : D0 22 AD F4 23 C9 03 D0  EA
2CD9 : 1B 20 D6 09 C5 52 20 48  DB
2CE1 : 41 54 20 45 54 57 41 53  A9
2CE9 : 20 5A 55 20 45 53 53 45  56
2CF1 : 4E 21 0D 00 60 20 D6 09  88
2CF9 : D3 49 45 20 53 45 48 45  D1
2D01 : 4E 20 45 49 4E 45 4E 20  62
2D09 : CB 4C 45 49 44 45 52 56  D9
2D11 : 45 52 4B 41 45 55 46 45  1D
2D19 : 52 21 0D 00 AD DB 23 C9  19
2D21 : 0A D0 2C AD F1 23 C9 03  B9
2D29 : D0 25 20 D6 09 C5 52 20  B7
2D31 : 48 41 54 20 45 49 4E 45  95
2D39 : 4E 20 C1 4E 5A 55 47 20  7F
2D41 : 5A 55 20 56 45 52 4B 41  AF
2D49 : 55 46 45 4E 21 0D 00 60  18
2D51 : 20 D6 09 D3 49 45 20 53  7F
2D59 : 45 48 45 4E 20 41 4E 20  63
2D61 : 53 49 43 48 20 48 45 52  31
2D69 : 41 42 2C 20 4B 4F 45 4E  BB
2D71 : 4E 45 4E 0D 41 42 45 52  77
2D79 : 20 4E 49 43 48 54 53 20  30
2D81 : C2 45 53 4F 4E 44 45 52  65
2D89 : 45 53 20 45 4E 54 44 45  4C
2D91 : 43 4B 45 4E 2E 0D 00 60  A1
2D99 : 20 D6 09 D3 49 45 20 53  C7
2DA1 : 45 48 45 4E 20 45 49 4E  13
2DA9 : 45 4E 20 D0 41 53 53 46  C0
2DB1 : 41 45 4C 53 43 48 45 52  43
2DB9 : 21 0D 00 AD D6 23 C9 0B  DA
2DC1 : D0 24 AD EC 23 C9 03 D0  DA
2DC9 : 1D 20 D6 09 C5 52 20 48  CD
2DD1 : 41 54 20 45 49 4E 45 4E  A6
2DD9 : 20 D0 41 53 53 20 55 45  32
2DE1 : 42 45 52 21 0D 00 60 20  11
2DE9 : D6 09 D3 49 45 20 53 45  8F
2DF1 : 48 45 4E 20 45 49 4E 45  D6
2DF9 : 4E 20 CC 41 4E 44 57 49  AA
2E01 : 52 54 2E 0D 00 AD DA 23  CA
2E09 : C9 0C D0 46 AD F0 23 C9  58
2E11 : 03 D0 3F 20 D6 09 C5 52  C2
2E19 : 20 48 41 54 20 C8 55 53  7C
2E21 : 54 45 4E 21 0D C6 55 45  B6
2E29 : 52 20 C8 55 53 54 45 4E  F2
2E31 : 53 41 46 54 20 57 55 45  DE
2E39 : 52 44 45 20 45 52 20 49  FD
2E41 : 48 4E 45 4E 20 C7 45 4C  B9
2E49 : 44 20 47 45 42 45 4E 21  E2
2E51 : 0D 00 60 20 D6 09 D3 49  12
2E59 : 45 20 53 45 48 45 4E 20  54
2E61 : 45 49 4E 45 4E 20 CD 4F  43
2E69 : 4E 54 45 55 52 20 44 45  9F
2E71 : 52 20 D7 41 53 53 45 52  7B
2E79 : 57 45 52 4B 45 2E 0D C5  F6
2E81 : 52 20 48 41 54 20 4E 49  30
2E89 : 43 48 54 53 20 C2 45 53  44
2E91 : 4F 4E 44 45 52 45 53 20  9E
2E99 : 42 45 49 20 53 49 43 48  F1
2EA1 : 2E 0D 00 60 20 D6 09 D3  E6
2EA9 : 49 45 20 53 45 48 45 4E  50
2EB1 : 20 44 45 4E 20 D0 4F 4C  6D
2EB9 : 49 5A 49 53 54 45 4E 20  D5
2EC1 : CD 45 59 45 52 21 0D 00  92
2EC9 : 60 20 D6 09 D3 49 45 20  ED
2ED1 : 53 45 48 45 4E 20 44 45  03
2ED9 : 4E 20 D0 4F 4C 49 5A 49  60
2EE1 : 53 54 45 4E 20 CD 55 45  CA
2EE9 : 4C 4C 45 52 2E 0D 00 60  03
2EF1 : 20 D6 09 D3 49 45 20 53  1F
2EF9 : 45 48 45 4E 20 45 49 4E  6B
2F01 : 45 4E 20 D3 54 52 45 49  6F
2F09 : 46 45 4E 50 4F 4C 49 5A  C1
2F11 : 49 53 54 45 4E 2C 20 20  C9
2F19 : 20 20 20 44 45 4E 20 53  C8
2F21 : 49 45 20 4E 49 43 48 54  57
2F29 : 20 4B 45 4E 4E 45 4E 21  95
2F31 : 0D 00 60 4C F1 2E 20 D6  9F
2F39 : 09 D3 49 45 20 53 45 48  69
2F41 : 45 4E 20 45 49 4E 45 4E  17
2F49 : 20 C6 4F 45 52 53 54 45  E4
2F51 : 52 2C 20 44 45 52 20 53  58
2F59 : 43 48 45 49 4E 42 41 52  DB
2F61 : 20 4A 45 4D 41 4E 44 45  C3
2F69 : 4E 20 53 55 43 48 54 21  51
2F71 : 0D 00 60 20 D6 09 D3 49  32
2F79 : 45 20 53 45 48 45 4E 20  74
2F81 : 45 49 4E 45 20 48 55 45  CB
2F89 : 42 53 43 48 45 20 D3 54  9C
2F91 : 45 57 41 52 44 45 53 53  7F
2F99 : 2E 0D D3 49 45 20 4C 41  75
2FA1 : 45 43 48 45 4C 54 20 53  D1
2FA9 : 49 45 20 41 4E 21 0D 00  E7
2FB1 : 60 20 D6 09 D6 4F 52 20  6A
2FB9 : 49 48 4E 45 4E 20 53 54  3E
2FC1 : 45 48 54 20 45 49 4E 45  A6
2FC9 : 20 D4 49 43 4B 45 54 56  EB
2FD1 : 45 52 4B 41 45 55 46 45  DD
2FD9 : 52 49 4E 2E 00 60 20 D6  5A
2FE1 : 09 D3 49 45 20 53 45 48  11
2FE9 : 45 4E 20 45 49 4E 45 4E  BF
2FF1 : 20 C6 4C 55 47 48 41 46  7A
2FF9 : 45 4E 50 4F 4C 49 5A 49  6E
3001 : 53 54 45 4E 2E 0D 00 60  A6
3009 : 4C DF 2F 20 D6 09 D3 49  AC
3011 : 45 20 53 45 48 45 4E 20  0C
3019 : 45 49 4E 45 4E 20 DA 4F  2F
3021 : 45 4C 4C 4E 45 52 20 42  55
3029 : 45 49 20 44 45 52 20 D0  AD
3031 : 41 53 53 2D 20 20 4B 4F  65
3039 : 4E 54 52 4F 4C 4C 45 21  AE
3041 : 0D 00 60 20 D6 09 D3 49  02
3049 : 45 20 53 45 48 45 4E 20  44
3051 : 45 49 4E 45 4E 20 D0 41  23
3059 : 53 53 41 4E 54 45 4E 2E  75
3061 : 0D 00 60 A2 18 BD 2B 24  3F
3069 : CD D3 23 D0 0B BD 44 24  FB
3071 : CD D4 23 D0 03 20 7D 30  13
3079 : CA D0 EA 60 BD 89 30 85  66
3081 : 14 BD A1 30 85 15 6C 14  BD
3089 : 00 51 41 82 A0 A3 A6 A9  E8
3091 : AC AF F6 99 E8 54 A5 CA  63
3099 : F1 34 37 74 B2 DF 09 0C  67
30A1 : 44 2D 2C 2C 2C 2C 2C 2C  3A
30A9 : 2C 2C 2C 2D 2D 2E 2E 2E  F5
30B1 : 2E 2F 2F 2F 2F 2F 30 30  B6
30B9 : 30 88 D0 0E 20 D6 09 45  8B
30C1 : 49 4E 45 4E 20 D0 41 53  81
30C9 : 53 00 88 D0 0C 20 D6 09  88
30D1 : 45 49 4E 45 20 D5 48 52  6E
30D9 : 00 88 D0 0E 20 D6 09 45  7B
30E1 : 49 4E 20 D4 49 43 4B 45  5A
30E9 : 54 00 88 D0 08 20 D6 09  68
30F1 : C7 45 4C 44 00 88 D0 14  A6
30F9 : 20 D6 09 45 54 57 41 53  1B
3101 : 20 C8 55 53 54 45 4E 53  94
3109 : 41 46 54 00 88 D0 0F 20  0E
3111 : D6 09 45 49 4E 45 4E 20  6F
3119 : C1 4E 5A 55 47 00 88 D0  7B
3121 : 08 20 D6 09 C7 4F 4C 44  C1
3129 : 00 88 D0 14 20 D6 09 45  8B
3131 : 49 4E 45 4E 20 D3 43 48  FB
3139 : 4C 55 45 53 53 45 4C 00  7C
3141 : 88 D0 0F 20 D6 09 45 54  6D
3149 : 57 41 53 20 C5 53 53 45  E8
3151 : 4E 00 88 D0 19 20 D6 09  DB
3159 : 45 49 4E 45 4E 20 DA 55  7B
3161 : 45 4E 44 53 43 48 4C 55  9B
3169 : 45 53 53 45 4C 00 88 D0  5E
3171 : 0C 20 D6 09 45 49 4E 45  C6
3179 : 20 C1 58 54 00 88 D0 0C  BA
3181 : 20 D6 09 45 49 4E 20 D3  27
3189 : 45 49 4C 00 88 D0 11 20  1A
3191 : D6 09 45 54 57 41 53 20  D5
3199 : D3 43 48 4D 55 43 4B 00  66
31A1 : 88 D0 10 20 D6 09 45 49  F7
31A9 : 4E 45 4E 20 C2 45 43 48  25
31B1 : 45 52 00 88 D0 13 20 D6  04
31B9 : 09 45 49 4E 45 4E 20 C2  4E
31C1 : 4C 45 49 53 54 49 46 54  BE
31C9 : 00 88 D0 0E 20 D6 09 45  6B
31D1 : 49 4E 45 4E 20 C2 41 4C  12
31D9 : 4C 00 88 D0 10 20 D6 09  D1
31E1 : 45 49 4E 45 20 DA 45 49  88
31E9 : 54 55 4E 47 00 88 D0 0F  0A
31F1 : 20 D6 09 45 49 4E 45 20  C4
31F9 : C2 52 49 4C 4C 45 00 88  C0
3201 : D0 0D 20 D6 09 45 49 4E  B7
3209 : 45 20 C4 4F 53 45 00 88  EA
3211 : D0 11 20 D6 09 45 49 4E  C9
3219 : 20 CB 4C 45 45 42 4C 41  F5
3221 : 54 54 00 88 D0 13 20 D6  84
3229 : 09 45 49 4E 45 20 C5 53  04
3231 : 53 45 4E 4D 41 52 4B 45  C2
3239 : 00 60 01 00 0D A0 15 AD  2F
3241 : 3B 32 D9 D5 23 D0 21 AD  5F
3249 : 3C 32 D9 EB 23 D0 19 8C  C8
3251 : 3D 32 A9 0D 20 D2 FF AC  A5
3259 : 3D 32 20 BA 30 A9 2C 20  50
3261 : D2 FF EE 09 0B AC 3D 32  7F
3269 : 88 D0 D4 AD 09 0B D0 1A  A4
3271 : 20 D6 09 0D 4E 49 43 48  AD
3279 : 54 53 20 C2 45 53 4F 4E  A0
3281 : 44 45 52 45 53 2E 0D 0D  9A
3289 : 00 60 20 D6 09 9D 2E 0D  ED
3291 : 0D 00 60 20 D6 09 C9 4E  34
3299 : 20 49 48 52 45 52 20 CE  BF
32A1 : 41 45 48 45 20 53 45 48  82
32A9 : 45 4E 20 53 49 45 20 00  C7
32B1 : A9 00 8D 09 0B AD D3 23  93
32B9 : 8D 3B 32 AD D4 23 8D 3C  3B
32C1 : 32 20 BC 34 4C 3E 32 20  79
32C9 : D6 09 45 49 4E 45 20 CB  C5
32D1 : 49 53 54 45 2C 20 44 49  E9
32D9 : 45 20 00 AD 1E 24 D0 14  53
32E1 : 20 D6 09 47 45 53 43 48  24
32E9 : 4C 4F 53 53 45 4E 20 49  F6
32F1 : 53 54 00 60 20 D6 09 4F  F6
32F9 : 46 46 45 4E 20 49 53 54  C0
3301 : 00 60 20 D6 09 45 49 4E  91
3309 : 45 4E 20 C2 4F 54 54 49  51
3311 : 43 48 00 60 AD 20 24 F0  D3
3319 : 1A 20 D6 09 45 49 4E 45  7C
3321 : 4E 20 4B 41 50 55 54 54  24
```

## Listing »Freiheit« (Fortsetzung)

```
3329 : 45 4E 20 CB 41 53 54 45  A1
3331 : 4E 00 60 20 D6 09 45 49  F9
3339 : 4E 45 4E 20 CB 41 53 54  7E
3341 : 45 4E 20 4D 49 54 20 C4  A0
3349 : 45 43 4B 45 4C 00 60 AD  4D
3351 : 21 24 D0 20 20 D6 09 45  24
3359 : 49 4E 45 4E 20 47 45 53  DC
3361 : 43 48 4C 4F 53 53 45 4E  47
3369 : 45 4E 20 D3 43 48 52 41  9A
3371 : 4E 4B 00 60 20 D6 09 45  D8
3379 : 49 4E 45 4E 20 4F 46 46  27
3381 : 45 4E 45 4E 20 D3 43 48  47
3389 : 52 41 4E 4B 00 60 20 D6  AA
3391 : 09 45 49 4E 45 4E 20 C2  26
3399 : 52 55 4E 4E 45 4E 00 60  7B
33A1 : 20 D6 09 45 49 4E 45 4E  D0
33A9 : 20 C2 45 48 41 45 4C 54  9D
33B1 : 45 52 00 60 20 D6 09 45  93
33B9 : 49 4E 20 D2 4F 48 52 00  0C
33C1 : 60 AD 25 24 D0 22 20 D6  12
33C9 : 09 45 49 4E 45 4E 20 47  67
33D1 : 45 53 43 48 4C 4F 53 53  CD
33D9 : 45 4E 45 4E 20 C3 4F 4E  5A
33E1 : 54 41 49 4E 45 52 00 60  9A
33E9 : 20 D6 09 45 49 4E 45 4E  18
33F1 : 20 4F 46 46 45 4E 45 4E  8C
33F9 : 20 C3 4F 4E 54 41 49 4E  AA
3401 : 45 52 00 60 20 D6 09 45  E3
3409 : 49 4E 45 20 C1 4C 41 52  F7
3411 : 4D 41 4E 4C 41 47 45 00  7F
3419 : 60 AD 27 24 D0 21 20 D6  E2
3421 : 09 45 49 4E 45 20 47 45  E6
3429 : 53 43 48 4C 4F 53 53 45  21
3431 : 4E 45 20 CD 55 45 4C 4C  2D
3439 : 54 4F 4E 4E 45 00 60 20  A8
3441 : D6 09 45 49 4E 45 20 4F  45
3449 : 46 46 45 4E 45 20 CD 55  05
3451 : 45 4C 4C 54 4F 4E 4E 45  85
3459 : 00 60 20 D6 09 45 49 4E  E9
3461 : 45 20 D4 4F 4E 4E 45 00  42
3469 : 60 20 D6 09 45 49 4E 20  C8
3471 : CD 4F 54 4F 52 42 4F 4F  F8
3479 : 54 00 60 AD 2A 24 D0 15  CC
3481 : 20 D6 09 45 49 4E 45 20  54
3489 : D3 54 45 49 4E 50 4C 41  1C
3491 : 54 54 45 00 60 20 D6 09  D5
3499 : 45 49 4E 45 20 D3 54 45  3C
34A1 : 49 4E 50 4C 41 54 54 45  42
34A9 : 20 4E 45 42 45 4E 20 45  5C
34B1 : 49 4E 45 4D 20 CC 4F 43  49
34B9 : 48 00 60 A0 0D AD 3B 32  BD
34C1 : D9 01 24 D0 21 AD 3C 32  13
34C9 : D9 0F 24 D0 19 8C 3D 32  9C
34D1 : A9 0D 20 D2 FF AC 3D 32  22
34D9 : 20 05 35 A9 2C 20 D2 FF  0D
34E1 : EE 09 0B AC 3D 32 88 D0  D5
34E9 : D4 60 C8 03 15 50 8F A1  D5
34F1 : B5 C2 05 1A 5B 6A 7C 32  EB
34F9 : 33 33 33 33 33 33 33 34  FB
3501 : 34 34 34 34 B9 EA 34 85  B2
3509 : 14 B9 F7 34 85 15 6C 14  59
3511 : 00 01 A9 00 BD 12 35 60  FB
3519 : AD 00 0A 0D 01 0A 0D 02  83
3521 : 0A 0D 04 0A 0D 03 0A 60  C6
3529 : 02 AD FB 09 C9 0D D0 37  D9
3531 : 20 19 35 0D FC 09 0D FD  15
3539 : 09 0D FE 09 0D FF 09 D0  40
3541 : 26 20 D6 09 D3 49 45 20  2B
3549 : 48 41 42 45 4E 20 00 A9  A4
3551 : 00 8D 09 0B A9 01 8D 3B  0B
3559 : 32 A9 03 8D 3C 32 20 3E  25
3561 : 32 A9 00 8D 12 35 60 AD  C1
3569 : FB 09 C9 1D F0 03 4C 82  5C
3571 : 36 20 19 35 0D FD 09 0D  A3
3579 : FE 09 0D FF 09 F0 03 4C  FC
3581 : 82 36 A9 00 8D 29 35 8D  9B
3589 : 12 35 A2 15 BD D5 23 C9  2C
3591 : 01 D0 0A BD EB 23 C9 03  3A
3599 : D0 03 EE 29 35 CA D0 EC  92
35A1 : AD 29 35 C9 05 B0 45 20  95
35A9 : 19 35 0D FE 09 0D FF 09  8B
35B1 : 0D FD 09 F0 06 A9 01 8D  EA
35B9 : 12 35 60 AE FC 09 BD D5  0E
35C1 : 23 CD D3 23 D0 45 BD EB  2A
35C9 : 23 CD D4 23 D0 3D A9 01  0C
35D1 : 9D D5 23 A9 03 9D EB 23  6A
35D9 : 20 D6 09 CF 4B 2C 20 47  C6
35E1 : 45 4E 4F 4D 4D 45 4E 2E  5F
35E9 : 0D 0D 00 60 20 D6 09 D3  0D
35F1 : 49 45 20 54 52 41 47 45  46
35F9 : 4E 20 5A 55 20 56 49 45  FD
3601 : 4C 21 0D 0D 00 A9 00 8D  2B
3609 : FA 09 60 BD D5 23 C9 01  F7
3611 : D0 0A BD EB 23 C9 03 D0  01
3619 : 03 4C 57 36 20 D6 09 C4  45
3621 : 45 52 20 C7 45 47 45 4E  D1
3629 : 53 54 41 4E 44 20 49 53  D2
3631 : 54 20 48 49 45 52 20 4E  D5
3639 : 49 43 48 54 20 5A 55 20  2B
3641 : 53 45 48 45 4E 21 20 20  A0
3649 : 20 20 20 20 20 20 0D 00  BD
3651 : A9 00 8D FA 09 60 20 D6  7F
3659 : 09 D3 49 45 20 48 41 42  15
3661 : 45 4E 20 44 49 45 53 45  F5
3669 : 4E 20 C7 45 47 45 4E 53  E0
3671 : 54 41 4E 44 20 42 45 52  50
3679 : 45 49 54 53 2E 0D 0D 00  62
3681 : 60 AD FB 09 C9 19 F0 03  07
3689 : 4C FF 36 20 19 35 0D FD  D2
3691 : 09 0D FE 09 0D FF 09 F0  D8
3699 : 03 EA EA 60 AE FC 09 BD  4A
36A1 : D5 23 C9 01 D0 2C BD EB  D8
36A9 : 23 C9 03 D0 25 AD D3 23  E1
36B1 : 9D D5 23 AD D4 23 9D EB  6C
36B9 : 23 20 D6 09 CF 4B 2C 20  0B
36C1 : 56 45 52 4C 4F 52 45 4E  11
36C9 : 2E 0D 0D 00 A9 00 8D 12  B6
36D1 : 35 60 20 D6 09 D3 49 45  F8
36D9 : 20 48 41 54 54 45 4E 20  E1
36E1 : 44 45 4E 20 C7 45 47 45  AE
36E9 : 4E 53 54 41 4E 44 20 4E  42
36F1 : 49 43 48 54 21 0D 0D 00  27
36F9 : A9 00 8D 12 35 60 AD FB  4D
3701 : 09 C9 11 F0 03 4C 67 37  F0
3709 : 20 19 35 0D FC 09 0D FD  ED
3711 : 09 0D FF 09 0D FE 09 F0  91
3719 : 01 60 20 D6 09 CE 45 55  F4
3721 : 45 53 20 D3 50 49 45 4C  8F
3729 : 20 4F 44 45 52 20 C1 42  5C
3731 : 42 52 55 43 48 20 44 45  7B
3739 : 53 20 D0 52 4F 47 52 41  16
3741 : 4D 4D 45 53 20 27 CE 27  B5
3749 : 20 42 5A 57 20 27 C1 27  9C
3751 : 20 3F 0D 0D 00 20 E4 FF  8A
3759 : F0 FB C9 41 D0 01 00 C9  8A
3761 : 4E D0 F2 4C 16 08 AD FB  AE
3769 : 09 C9 12 D0 31 20 19 35  D8
3771 : 0D FC 09 0D FD 09 0D FE  3B
3779 : 09 0D FF 09 F0 01 60 20  03
3781 : D6 09 D3 49 45 20 57 41  2F
3789 : 52 54 45 4E 20 45 49 4E  0E
3791 : 45 20 D7 45 49 4C 45 2E  ED
3799 : 0D 0D 00 4C 13 35 AD FB  40
37A1 : 09 C9 10 D0 5D 20 19 35  53
37A9 : 0D FC 09 0D FD 09 0D FE  73
37B1 : 09 0D FF 09 F0 01 60 20  3B
37B9 : D6 09 C4 41 53 20 D3 50  93
37C1 : 49 45 4C 20 49 53 54 20  85
37C9 : 55 4E 54 45 52 42 52 4F  22
37D1 : 43 48 45 4E 20 42 49 53  33
37D9 : 20 53 49 45 20 44 49 45  72
37E1 : 20 20 CC 45 45 52 54 41  A8
37E9 : 53 54 45 20 44 52 55 45  72
37F1 : 43 4B 45 4E 21 0D 0D 00  A4
37F9 : 20 E4 FF C9 20 D0 F9 4C  CD
3801 : 13 35 AD FB 09 C9 0E F0  93
3809 : 07 C9 0F F0 03 4C 44 39  ED
3811 : 20 19 35 0D FC 09 0D FD  F5
3819 : 09 0D FE 09 0D FF 09 F0  60
3821 : 01 60 20 D6 09 C4 49 53  BB
3829 : 4B 45 54 54 45 20 4F 44  D2
3831 : 45 52 20 CB 41 53 53 45  A7
3839 : 54 54 45 20 20 20 20 20  D1
3841 : 20 20 20 20 20 20 20 20  41
3849 : 20 20 20 20 20 27 C4 27  22
3851 : 20 42 5A 57 2E 20 27 CB  2C
3859 : 27 20 3F 0D 00 20 E4 FF  96
3861 : F0 FB C9 44 F0 07 C9 4B  4F
3869 : F0 1A 4C 5E 38 20 D6 09  37
3871 : 2D 3E 20 C4 49 53 4B 45  45
3879 : 54 54 45 0D 00 A9 08 85  63
3881 : BA 4C 99 38 20 D6 09 2D  06
3889 : 3E 20 CB 41 53 53 45 54  80
3891 : 54 45 0D 00 A9 01 85 BA  F9
3899 : A9 01 85 B9 85 B8 A9 00  20
38A1 : 85 BB A9 02 85 BC A9 04  9B
38A9 : 85 B7 A9 40 8D 00 02 A9  B0
38B1 : 3A 8D 01 02 A9 20 8D 02  08
38B9 : 02 20 D6 09 D3 50 49 45  11
38C1 : 4C 4B 45 4E 4E 55 4E 47  25
38C9 : 20 C1 2D DA 20 3F 20 00  ED
38D1 : 20 E4 FF F0 FB C9 41 90  B5
38D9 : F7 C9 5B B0 F3 8D 03 02  5D
38E1 : 20 D2 FF 20 D6 09 0D 0D  72
38E9 : 00 AD FB 09 C9 0F F0 22  FD
38F1 : A9 00 A2 D3 A0 23 20 D9  15
38F9 : 4A B0 32 A9 0D 20 D2 FF  7A
3901 : A9 00 8D 12 35 68 68 A9  DC
3909 : 00 8D FA 09 8D FB 09 4C  25
3911 : D2 29 A2 90 A0 24 A9 D3  AC
3919 : 85 FB A9 23 85 FC A9 FB  49
3921 : 20 E7 4A B0 08 A9 0D 20  20
3929 : D2 FF 4C 13 35 20 D6 09  32
3931 : 0D C9 2F CF 20 C5 52 52  07
3939 : 4F 52 21 21 21 0D 0D 00  CD
3941 : 4C 13 35 AD FB 09 C9 13  6F
3949 : F0 03 4C 03 3A 20 19 35  A2
3951 : 0D FC 09 0D FD 09 0D FE  1B
3959 : 09 0D FF 09 F0 01 60 20  E3
3961 : D6 09 C6 4F 4C 47 45 4E  08
3969 : 44 45 20 D6 45 52 42 45  AD
3971 : 4E 20 42 5A 57 2E 20 C1  96
3979 : 42 4B 55 45 52 5A 55 4E  49
3981 : 47 45 4E 20 53 49 4E 44  44
3989 : 20 20 5A 55 4C 41 45 53  85
3991 : 53 49 47 3A 20 20 20 20  66
3999 : 20 20 20 20 0D 00 EA 20  92
39A1 : 3A 12 20 43 12 8D 52 12  50
39A9 : 20 43 12 8D 53 12 20 43  6E
39B1 : 12 8D 54 12 AD 52 12 D0  39
39B9 : 03 4C EE 39 AD 53 12 C9  16
39C1 : 02 F0 0C AE 52 12 20 43  D1
39C9 : 12 CA D0 FA 4C A3 39 A9  EE
39D1 : 0D 20 D2 FF AE 52 12 20  A9
39D9 : 43 12 29 7F 20 D2 FF CA  8E
39E1 : D0 F5 A9 2C 20 D2 FF 20  75
39E9 : 09 3A 4C A3 39 A9 14 20  08
39F1 : D2 FF A9 2E 20 D2 FF A9  DF
39F9 : 0D 20 D2 FF 20 D2 FF 4C  FC
3A01 : 13 35 AD FB 09 4C 18 3A  61
3A09 : A0 00 84 FB 84 FC C6 FB  8D
3A11 : D0 FC C6 FC D0 F8 60 C9  9A
3A19 : 1B F0 03 4C 3F 3B 20 19  77
3A21 : 35 0D FE 09 0D FD 09 F0  84
3A29 : 01 60 AE FF 09 BD 2B 24  79
3A31 : CD D3 23 D0 08 BD 44 24  93
3A39 : D0 03 4C 6A 3A 20 D6 09  FD
3A41 : D3 49 45 20 53 45 48 45  19
3A49 : 4E 20 44 49 45 53 45 20  26
3A51 : D0 45 52 53 4F 4E 20 48  3B
3A59 : 49 45 52 20 4E 49 43 48  AA
3A61 : 54 21 20 0D 0D 00 4C 13  18
3A69 : 35 AC FC 09 B9 D5 23 C9  BF
3A71 : 01 D0 0A B9 EB 23 C9 03  99
3A79 : D0 03 4C AB 3A 20 D6 09  65
3A81 : D3 49 45 20 48 41 42 45  70
3A89 : 4E 20 44 49 45 53 45 4E  C2
3A91 : 20 C7 45 47 45 4E 53 54  8C
3A99 : 41 4E 44 20 4E 49 43 48  E3
3AA1 : 54 21 20 20 0D 0D 00 4C  64
3AA9 : 13 35 98 DD 26 3B D0 46  45
3AB1 : 8A 99 D5 23 A9 03 99 EB  D3
3AB9 : 23 20 D6 09 CF 4B 2E 20  13
3AC1 : C4 49 45 20 D0 45 52 53  A6
3AC9 : 4F 4E 20 48 41 54 20 44  10
3AD1 : 45 4E 20 C7 45 47 45 4E  7F
3AD9 : 53 54 41 4E 44 20 20 20  77
3AE1 : 20 20 20 20 47 45 4E 4F  94
3AE9 : 4D 4D 45 4E 2E 20 20 20  9D
3AF1 : 0D 0D 00 4C 13 35 20 D6  18
3AF9 : 09 C4 49 45 20 D0 45 52  A2
3B01 : 53 4F 4E 20 57 49 4C 4C  1D
3B09 : 20 44 45 4E 20 C7 45 47  4A
3B11 : 45 4E 53 54 41 4E 44 20  B5
3B19 : 4E 49 43 48 54 21 20 20  F5
3B21 : 20 0D 00 4C 13 35 00 00  2C
3B29 : 00 00 00 00 00 00 15 07  8C
3B31 : 02 05 00 00 00 00 00 00  B6
3B39 : 03 04 00 00 01 00 AD FB  FD
3B41 : 09 C9 14 F0 03 4C B2 3B  26
3B49 : AD FC 09 0D FD 09 0D FE  B3
3B51 : 09 F0 01 60 AE FF 09 D0  CF
3B59 : 27 20 D6 09 D3 49 45 20  44
3B61 : 52 45 44 45 4E 20 4D 49  BD
3B69 : 54 20 53 49 43 48 20 53  69
3B71 : 45 4C 42 53 54 21 20 20  E6
3B79 : 20 20 0D 0D 00 4C 13 35  A7
3B81 : BD 2B 24 CD D3 23 D0 D1  D4
3B89 : BD 44 24 D0 CC EA EA EA  31
3B91 : 20 D6 09 C4 49 45 20 D0  D8
3B99 : 45 52 53 4F 4E 20 48 4F  6C
3BA1 : 45 52 54 20 5A 55 2E 20  72
3BA9 : 20 20 20 0D 0D 00 4C 13  AB
3BB1 : 35 AD FB 09 C9 15 F0 03  EC
3BB9 : 4C EE 3C 20 19 35 0D FD  FB
3BC1 : 09 0D FE 09 F0 01 60 AE  2B
3BC9 : FC 09 D0 06 AE FF 09 D0  EF
3BD1 : 04 60 4C 5F 3C BD 2B 24  AB
3BD9 : CD D3 23 D0 08 BD 44 24  3B
3BE1 : D0 03 4C 1C 3C EA EA EA  66
3BE9 : 20 D6 09 C4 49 45 20 D0  30
3BF1 : 45 52 53 4F 4E 20 49 53  D0
3BF9 : 54 20 4E 49 43 48 54 20  22
3C01 : 49 4E 20 49 48 52 45 52  73
3C09 : 20 CE 41 45 48 45 21 20  FD
3C11 : 20 20 20 20 20 20 0D 00  85
3C19 : 4C 13 35 EA EA EA 20 D6  CE
3C21 : 09 C1 4E 20 44 45 52 20  9A
3C29 : D0 45 52 53 4F 4E 20 49  15
3C31 : 53 54 20 4E 49 43 48 54  F9
3C39 : 53 20 C2 45 53 4F 4E 44  67
3C41 : 45 52 45 53 20 20 20 20  2F
3C49 : 20 5A 55 20 53 45 48 45  FB
3C51 : 4E 2E 20 20 20 20 20 20  86
3C59 : 0D 0D 00 4C 13 35 BD D5  F4
```

```
3C61 : 23 CD D3 23 D0 08 BD EB   E0
3C69 : 23 D0 03 4C B5 3C 4C F4   97
3C71 : 3C 20 D6 09 C4 45 52 20   94
3C79 : C7 45 47 45 4E 53 54 41   B1
3C81 : 4E 44 20 49 53 54 20 41   FD
3C89 : 55 53 53 45 52 48 41 4C   0A
3C91 : 42 20 49 48 52 45 53 20   1C
3C99 : 20 20 20 20 C2 4C 49 43   10
3CA1 : 4B 46 45 4C 44 45 53 21   E8
3CA9 : 20 20 20 20 20 20 0D 0D   37
3CB1 : 00 4C 13 35 EA EA EA 20   35
3CB9 : D6 09 C4 45 52 20 C7 45   BD
3CC1 : 47 45 4E 53 54 41 4E 44   BA
3CC9 : 20 48 41 54 20 4B 45 49   EC
3CD1 : 4E 45 20 C2 45 53 4F 4E   EB
3CD9 : 44 45 52 48 45 49 54 45   D8
3CE1 : 4E 21 20 20 20 20 20 20   90
3CE9 : 0D 00 4C 13 35 AD FB 09   2F
3CF1 : 4C 08 3D BD D5 23 C9 01   E8
3CF9 : D0 0A BD EB 23 C9 03 D0   E9
3D01 : 03 4C B5 3C 4C 72 3C C9   FC
3D09 : 1A F0 03 4C 98 3D 20 19   0C
3D11 : 35 0D FD 09 0D FE 09 0D   74
3D19 : FF 09 F0 01 60 AE FC 09   7B
3D21 : F0 05 E0 09 F0 01 60 A2   CB
3D29 : 09 BD D5 23 CD D3 23 D0   94
3D31 : 08 BD EB 23 D0 03 4C 48   5E
3D39 : 3D BD D5 23 C9 01 D0 2A   6B
3D41 : BD EB 23 C9 03 D0 23 EA   0F
3D49 : EA EA A9 09 9D D5 23 A9   9C
3D51 : 03 9D EB 23 20 D6 09 D3   07
3D59 : 49 45 20 45 53 53 45 4E   77
3D61 : 2E 20 20 20 0D 0D 00 4C   7D
3D69 : 2F 4E EA EA EA 20 D6 09   F4
3D71 : D3 49 45 20 48 41 42 45   60
3D79 : 4E 20 4E 49 43 48 54 53   03
3D81 : 20 C5 53 53 42 41 52 45   C5
3D89 : 53 21 20 20 20 20 20 20   3D
3D91 : 20 0D 0D 00 4C 13 35 AD   09
3D99 : FB 09 EA EA EA EA 60 EA   8E
3DA1 : EA EA EA EA EA EA EA EA   A0
3DA9 : EA EA EA AA BD 19 29 D0 03   8F
3DB1 : 4C 33 2A 8D D3 23 A9 00   D0
3DB9 : 8D 12 35 20 8F 24 20 24   84
3DC1 : 29 20 00 3E 20 64 2A 20   D0
3DC9 : 94 32 20 64 30 EA EA EA   E7
3DD1 : 60 EA EA 4C 2A 35 AD FB   E5
3DD9 : 09 C9 01 90 F6 C9 0B B0   64
3DE1 : F2 8D FD 3D 20 19 35 0D   7B
3DE9 : FC 09 0D FD 09 0D FE 09   74
3DF1 : 0D FF 09 F0 01 60 AD FD   24
3DF9 : 3D 4C A0 3D 03 EA EA EA   35
3E01 : EA EA AD D4 23 F0 03 EA   02
3E09 : EA 60 AD D3 23 C9 01 D0   2F
3E11 : 11 A9 00 8D 1D 29 A2 0D   C9
3E19 : BD 1D 24 F0 05 A9 02 8D   4D
3E21 : 23 29 AD D3 23 C9 0B D0   0D
3E29 : 1A A9 00 8D 1D 29 A2 0C   B7
3E31 : BD D5 23 C9 01 D0 0C BD   1D
3E39 : EB 23 C9 03 D0 05 A9 0D   7F
3E41 : 8D 22 29 AD D3 23 C9 0D   77
3E49 : D0 0A A9 00 8D 1D 29 A9   42
3E51 : 0E 8D 23 29 AD D3 23 C9   AD
3E59 : 20 D0 05 A9 00 8D 1A 29   7F
3E61 : AD D3 23 C9 24 D0 05 A9   2A
3E69 : 00 8D 1B 29 AD D3 23 C9   B5
3E71 : 60 D0 0A A9 00 A2 0A 9D   6A
3E79 : 19 29 CA D0 FA EA EA 60   67
3E81 : 20 D6 09 11 11 11 11 11   71
3E89 : 11 C8 45 52 5A 4C 49 43   4E
3E91 : 48 45 4E 20 C7 4C 55 45   D2
3E99 : 43 4B 57 55 4E 53 43 48   1F
3EA1 : 21 0D 0D D3 49 45 20 48   D6
3EA9 : 41 42 45 4E 20 49 48 52   39
3EB1 : 45 20 57 4F 48 4C 56 45   91
3EB9 : 52 44 49 45 4E 54 45 20   05
3EC1 : C6 52 45 49 48 45 49 54   A7
3EC9 : 20 20 20 5A 55 52 55 45   14
3ED1 : 43 4B 47 45 57 4F 4E 4E   FA
3ED9 : 45 4E 21 0D 0D 0D 00 4C   01
3EE1 : 1B 37 20 D6 09 11 11 11   FA
3EE9 : 11 11 11 2E 2E 2E 20 44   EA
3EF1 : 41 53 20 54 41 54 20 57   54
3EF9 : 45 48 21 0D 11 D3 49 45   AC
3F01 : 20 53 49 4E 44 20 41 55   DC
3F09 : 53 20 49 48 52 45 52 20   A1
3F11 : D0 52 49 54 53 43 48 45   E2
3F19 : 20 47 45 46 41 4C 4C 45   29
3F21 : 4E 2E 0D C9 48 52 45 20   6F
3F29 : C6 4C 55 43 48 54 20 48   0B
3F31 : 41 42 45 4E 20 53 49 45   FB
3F39 : 20 4C 45 49 44 45 52 20   F2
3F41 : 4E 55 52 20 20 20 20 20   96
3F49 : 20 20 20 47 45 54 52 41   2D
3F51 : 45 55 4D 54 20 2D 20 45   95
3F59 : 49 4E 20 D7 55 4E 53 43   68
3F61 : 48 54 52 41 55 4D 2E 0D   23
3F69 : 0D D6 45 52 53 55 43 48   FA
3F71 : 45 4E 20 53 49 45 20 45   1A
3F79 : 53 20 44 4F 43 48 20 45   59
3F81 : 49 4E 4D 41 4C 20 45 52   EC
3F89 : 4E 53 54 48 41 46 54 21   79
3F91 : 21 0D 0D 0D 00 4C 1B 37   5B
3F99 : 20 D6 09 93 20 20 20 20   9D
3FA1 : 20 20 20 20 20 20 20 20   A1
3FA9 : 20 20 20 20 20 C6 D2 C5 C9   93
3FB1 : C8 C5 C9 D4 0D 0D 0D D3   7E
3FB9 : 49 45 20 42 45 46 49 4E   3E
3FC1 : 44 45 4E 20 53 49 43 48   5D
3FC9 : 20 49 4E 20 45 49 4E 45   88
3FD1 : 4D 20 C7 45 46 41 45 4E   E9
3FD9 : 47 4E 49 53 2E 0D CD 41   09
3FE1 : 4E 20 48 41 54 20 53 49   A0
3FE9 : 45 20 5A 55 20 4C 45 42   7E
3FF1 : 45 4E 53 4C 41 4E 47 45   EA
3FF9 : 52 20 C8 41 46 54 20 20   7D
4001 : 20 20 20 20 20 20 56 45   24
4009 : 52 55 52 54 45 49 4C 54   9D
4011 : 2C 20 57 45 49 4C 20 53   EA
4019 : 49 45 20 44 45 4D 20 CB   6C
4021 : 4F 45 4E 49 47 20 4E 49   3D
4029 : 4E 0D 27 C6 4F 4C 54 45   D4
4031 : 52 4F 4E 49 45 4E 27 20   BB
4039 : 41 55 46 20 44 45 52 20   B2
4041 : D3 54 52 41 53 53 45 20   20
4049 : 49 4E 53 20 20 20 20 20   56
4051 : 20 20 C7 45 53 49 43 48   39
4059 : 54 20 47 45 53 45 48 45   43
4061 : 4E 20 48 41 42 45 4E 21   C3
4069 : 0D 11 11 11 11 11 C1 4C   9F
4071 : 53 20 C5 52 4C 45 49 43   2B
4079 : 48 54 45 52 55 4E 47 20   AC
4081 : 42 49 45 54 45 54 20 44   44
4089 : 41 53 20 D3 50 49 45 4C   F3
4091 : 20 20 20 20 20 20 44 49   74
4099 : 45 20 D7 4F 45 52 54 45   91
40A1 : 52 20 27 C8 C9 CC C6 C5   90
40A9 : 27 0D 55 4E 44 20 27 C2   DD
40B1 : C5 C6 C5 C8 CC C5 27 20   3C
40B9 : 21 21 21 0D 11 11 11 11   55
40C1 : 11 C4 52 55 45 43 4B 45   9A
40C9 : 4E 20 53 49 45 20 42 49   16
40D1 : 54 54 45 20 45 49 4E 45   07
40D9 : 20 D4 41 53 54 45 21 21   54
40E1 : 21 20 20 20 20 20 20 20   E2
40E9 : 20 0D 0D 00 EA EA EA 20   C5
40F1 : E4 FF F0 FB A9 93 20 D2   EE
40F9 : FF EA EA EA EA [illegible] 60 AD   68
4101 : D3 23 C9 01 F0 03 4C 84   5A
4109 : 42 AD FB 09 C9 0C F0 0D   1D
4111 : C9 15 F0 68 C9 16 F0 67   8E
4119 : C9 1E F0 66 60 20 19 35   D0
4121 : 0D FC 09 0D FD 09 0D FE   EB
4129 : 09 0D FF 09 F0 01 60 20   B3
4131 : D6 09 DA 55 52 20 DA 45   09
4139 : 49 54 20 4B 41 4E 4E 20   1E
4141 : 49 43 48 20 41 4C 53 20   46
4149 : C8 49 4C 46 45 20 4E 55   CB
4151 : 52 20 41 55 46 20 44 41   A7
4159 : 53 20 D7 4F 52 54 20 27   33
4161 : C2 C5 C6 C5 C8 CC C5 27   C8
4169 : 20 48 49 4E 57 45 49 53   35
4171 : 45 4E 2E 20 20 20 0D 0D   BE
4179 : 00 4C 13 35 4C 86 41 4C   A1
4181 : E3 41 4C 20 42 20 19 35   10
4189 : 0D FD 09 0D FC 09 0D FF   C5
4191 : 09 F0 01 60 AD FE 09 D0   F7
4199 : 03 4C B7 3D C9 0D D0 F3   88
41A1 : EA 20 D6 09 D3 49 45 20   4F
41A9 : 4B 4F 45 4E 4E 45 4E 20   40
41B1 : 4E 49 43 48 54 53 20 C2   64
41B9 : 45 53 4F 4E 44 45 52 45   88
41C1 : 53 20 20 20 20 20 20 20   F4
41C9 : 20 20 20 20 46 45 53 54   8A
41D1 : 53 54 45 4C 4C 45 4E 21   94
41D9 : 20 20 20 20 0D 0D 00 4C   E7
41E1 : 13 35 20 19 35 0D FC 09   7C
41E9 : 0D FD 09 0D FE 09 0D FF   45
41F1 : 09 F0 01 60 EA EA EA 20   B1
41F9 : D6 09 D3 49 45 20 48 4F   87
4201 : 45 52 45 4E 20 4E 49 43   AB
4209 : 48 54 53 20 C2 45 53 4F   96
4211 : 4E 44 45 52 45 53 21 20   D1
4219 : 20 0D 0D 00 4C 13 35 20   76
4221 : 19 35 0D FC 09 0D FD 09   BB
4229 : 0D FF 09 F0 01 60 AD FE   5E
4231 : 09 C9 0D D0 F8 EA EA EA   E5
4239 : 20 D6 09 D3 49 45 20 48   51
4241 : 45 42 45 4E 20 44 49 45   96
4249 : 20 D3 54 45 49 4E 50 4C   F1
4251 : 41 54 54 45 20 55 4E 44   E9
4259 : 20 4C 45 47 45 4E 20 53   C8
4261 : 49 45 20 57 4F 41 4E 44   01
4269 : 45 52 53 20 41 42 2E 20   CF
4271 : 20 0D 0D 00 A9 01 A2 0D   A2
4279 : 9D 1D 24 A9 02 8D 23 29   4F
4281 : 4C 13 35 C9 02 F0 03 4C   2A
4289 : E2 43 AD FB 09 C9 0C F0   E9
4291 : 09 C9 15 F0 0B C9 16 F0   EB
4299 : 07 60 4C A4 42 4C D8 42   E6
42A1 : 4C ED 42 20 19 35 0D FC   E2
42A9 : 09 0D FD 09 0D FE 09 0D   E0
42B1 : FF 09 F0 01 60 EA EA EA   70
42B9 : 20 D6 09 CE 55 54 5A 45   4C
42C1 : 4E 20 53 49 45 20 49 48   28
42C9 : 52 45 20 D3 49 4E 4E 45   0B
42D1 : 21 0D 0D 00 4C 13 35 20   2F
42D9 : 19 35 0D FC 09 0D FD 09   73
42E1 : 0D FE 09 0D FF 09 F0 01   5F
42E9 : 60 4C B7 3D EA EA EA 20   F7
42F1 : 19 35 0D FC 09 0D FD 09   8B
42F9 : 0D FE 09 0D FF 09 F0 01   77
4301 : 60 A9 00 8D 09 0B AD 2E   E4
4309 : 24 C9 04 D0 2B EE 09 0B   91
4311 : 20 D6 09 C9 4D 20 D7 45   D8
4319 : 53 54 45 4E 20 48 4F 45   BE
4321 : 52 45 4E 20 53 49 45 20   82
4329 : 45 49 4E 20 C7 45 52 41   1D
4331 : 45 55 53 43 48 2E 0D 00   88
4339 : AD 2E 24 C9 05 D0 26 EE   8D
4341 : 09 0B 20 D6 09 C9 4D 20   07
4349 : CE 4F 52 44 45 4E 20 48   B4
4351 : 4F 45 52 45 4E 20 53 49   46
4359 : 45 20 D3 43 48 52 49 54   F0
4361 : 54 45 2E 0D 00 AD 2F 24   F8
4369 : C9 05 D0 1D EE 09 0B 20   30
4371 : D6 09 C9 4D 20 CE 4F 52   42
4379 : 44 45 4E 20 49 53 54 20   B8
4381 : 4A 45 4D 41 4E 44 2E 0D   C3
4389 : 00 AD 2F 24 C9 03 D0 22   EC
4391 : EE 09 0B 20 D6 09 C9 4D   42
4399 : 20 CF 53 54 45 4E 20 48   D8
43A1 : 4F 45 52 45 4E 20 53 49   96
43A9 : 45 20 45 54 57 41 53 2E   03
43B1 : 0D 00 AD 09 0B D0 22 20   4B
43B9 : D6 09 D3 49 45 20 48 4F   47
43C1 : 45 52 45 4E 20 4E 49 43   6B
43C9 : 48 54 53 20 C2 45 53 4F   56
43D1 : 4E 44 45 52 45 53 2E 0D   9F
43D9 : 00 A9 0D 20 D2 FF 4C 13   7A
43E1 : 35 C9 03 F0 03 4C 56 44   4E
43E9 : AD FB 09 C9 0C D0 03 4C   FB
43F1 : 1E 41 C9 15 D0 03 4C D8   CD
43F9 : 42 C9 16 F0 01 60 AD 2F   EC
4401 : 24 C9 07 D0 2F 20 D6 09   47
4409 : C1 55 53 20 D2 49 43 48   63
4411 : 54 55 4E 47 20 CE 4F 52   E7
4419 : 44 45 4E 20 48 41 4C 41   F0
4421 : 45 4E 20 D3 43 48 52 49   62
4429 : 54 54 45 2E 20 20 0D 0D   10
4431 : 00 4C 13 35 20 D6 09 C5   2B
4439 : 53 20 49 53 54 20 52 45   73
4441 : 43 48 54 20 53 54 49 4C   57
4449 : 4C 20 48 49 45 52 21 0D   66
4451 : 0D 00 4C 13 35 C9 04 F0   67
4459 : 03 4C D3 44 AD FB 09 C9   72
4461 : 0C D0 03 4C 1E 41 C9 15   5D
4469 : D0 03 4C D8 42 C9 16 F0   96
4471 : 01 60 EA EA EA 20 19 35   39
4479 : 0D FC 09 0D FD 09 0D FE   43
4481 : 09 0D FF 09 F0 01 60 AD   24
4489 : 2E 24 C9 06 D0 25 EA EA   B4
4491 : EA 20 D6 09 C9 4D 20 CE   87
4499 : 4F 52 44 45 4E 20 49 53   7D
44A1 : 54 20 4A 45 4D 41 4E 44   E1
44A9 : 2E 20 20 20 20 0D 0D 00   92
44B1 : 4C 13 35 20 D6 09 CE 49   5C
44B9 : 43 48 54 53 20 52 45 47   D8
44C1 : 54 20 53 49 43 48 2E 20   93
44C9 : 20 20 20 20 0D 0D 00 4C   D7
44D1 : 13 35 C9 05 F0 03 4C 8C   03
44D9 : 45 AD FB 09 C9 0C D0 03   5B
44E1 : 4C 1E 41 C9 15 D0 03 4C   42
44E9 : D8 42 C9 16 F0 01 60 20   F0
44F1 : AF 45 AD 2F 24 C9 06 D0   DE
44F9 : 2B 20 D6 09 D6 4F 4E 20   6C
4501 : D7 45 53 54 45 4E 20 48   B2
4509 : 4F 45 52 54 20 4D 41 4E   28
4511 : 20 D3 43 48 52 49 54 54   5E
4519 : 45 2E 20 20 20 20 0D 0D   D3
4521 : 00 4C 13 35 C9 07 D0 2A   1F
4529 : 20 D6 09 D6 4F 4E 20 CF   59
4531 : 53 54 45 4E 20 48 4F 45   D6
4539 : 52 54 20 4D 41 4E 20 D3   16
4541 : 43 48 52 49 54 54 45 2E   BF
4549 : 20 20 20 20 0D 0D 00 4C   57
4551 : 13 35 EA EA EA 20 D6 09   34
4559 : CD 41 4E 20 48 4F 45 52   17
4561 : 54 20 D7 41 53 53 45 52   6D
4569 : 20 56 4F 4E 20 44 45 52   30
4571 : 20 C4 45 43 4B 45 20 54   B5
4579 : 52 4F 50 46 45 4E 2E 20   0F
4581 : 20 20 20 20 20 20 0D 00   F5
4589 : 4C 13 35 C9 06 F0 03 4C   72
4591 : 5F 46 AD FB 09 C9 0C D0   AF
4599 : 03 4C 1E 41 C9 15 D0 03   01
45A1 : 4C D8 42 C9 16 F0 01 60   D1
```

### Listing »Freiheit« (Fortsetzung)

```
45A9 : EA EA EA 4C C4 45 20 19   76
45B1 : 35 0D FC 09 0D FD 09 0D   CC
45B9 : FE 09 0D FF 09 D0 01 60   5B
45C1 : 68 68 60 20 AF 45 AD 2E   B2
45C9 : 24 C9 06 D0 36 20 D6 09   3F
45D1 : D3 49 45 20 48 4F 45 52   57
45D9 : 45 4E 20 4E 55 52 20 44   0B
45E1 : 45 4E 20 D7 41 45 52 54   B1
45E9 : 45 52 20 4E 45 42 45 4E   41
45F1 : 20 49 48 4E 45 4E 2E 20   51
45F9 : 20 20 20 20 20 20 0D 00   6D
4601 : 4C 13 35 C9 09 D0 10 20   F5
4609 : D6 09 D6 4F 4E 20 4F 42   AB
4611 : 45 4E 20 00 4C 40 46 C9   F9
4619 : 05 D0 11 20 D6 09 D6 4F   7E
4621 : 4E 20 CF 53 54 45 4E 20   C7
4629 : 00 4C 40 46 C9 04 D0 0F   46
4631 : 20 D6 09 D6 4F 4E 20 D3   69
4639 : 55 45 44 45 4E 20 00 20   11
4641 : D6 09 48 4F 45 52 45 4E   30
4649 : 20 53 49 45 20 C7 45 52   08
4651 : 41 45 55 53 43 48 45 2E   DD
4659 : 0D 0D 00 4C 13 35 C9 07   87
4661 : F0 03 4C 00 4B AD FB 09   0A
4669 : C9 0C D0 03 4C 1E 41 C9   1B
4671 : 15 D0 03 4C 9D 46 C9 16   9B
4679 : D0 03 4C DD 47 C9 18 D0   5E
4681 : 03 4C C0 48 C9 19 D0 03   92
4689 : 4C A9 49 C9 1C D0 03 4C   22
4691 : 35 49 C9 1D D0 03 4C 3D   52
4699 : 4A EA EA 60 EA EA EA 20   11
46A1 : 19 35 0D FD 09 0D FD 09   5B
46A9 : 0D FF 09 F0 01 60 AD FE   DE
46B1 : 09 C9 04 F0 03 4C DB 42   38
46B9 : 4C 4F 47 EA AD 21 24 D0   F2
46C1 : 36 EA EA EA 20 D6 09 D3   09
46C9 : 49 45 20 53 45 48 45 4E   70
46D1 : 20 45 49 4E 45 4E 20 47   B6
46D9 : 45 53 43 48 4C 4F 53 53   D5
46E1 : 45 4E 45 4E 20 D3 43 48   A7
46E9 : 52 41 4E 4B 2E 20 20 20   7E
46F1 : 20 20 0D 00 4C 13 35 EA   6D
46F9 : EA EA 20 D6 09 D3 49 45   1A
4701 : 20 53 45 48 45 4E 20 45   F7
4709 : 49 4E 45 4E 20 4F 46 46   B7
4711 : 45 4E 45 4E 20 D3 43 48   D7
4719 : 52 41 4E 4B 2E 20 20 20   AE
4721 : 20 20 20 0D C4 41 52 49   2D
4729 : 4E 20 53 45 48 45 4E 20   2D
4731 : 53 49 45 3A 20 00 A9 00   6A
4739 : 8D 09 0B A9 04 8D 3B 32   41
4741 : A9 01 8D 3C 32 20 3E 32   D7
4749 : EA EA EA 4C 13 35 AE FC   7C
4751 : 09 D0 03 4C BC 46 EA 4C   4F
4759 : 73 4B C9 F0 C9 04 4C A6   0A
4761 : 47 BD EB 23 C9 01 F0 03   54
4769 : 4C A6 47 EA EA EA 20 D6   6C
4771 : 09 C1 4E 20 44 45 4D 20   D6
4779 : C7 45 47 45 4E 53 54 41   B1
4781 : 4E 44 20 49 53 54 20 4E   1B
4789 : 49 43 48 54 53 20 C2 45   DC
4791 : 53 4F 4E 44 45 52 45 53   4B
4799 : 2E 20 20 20 20 20 20 20   A7
47A1 : 0D 00 4C 13 35 EA EA EA   50
47A9 : 20 D6 09 C4 45 52 20 C7   06
47B1 : 45 47 45 4E 53 54 41 4E   2E
47B9 : 44 20 49 53 54 20 4E 49   DC
47C1 : 43 48 54 20 49 4D 20 D3   69
47C9 : 43 48 52 41 4E 4B 21 20   F1
47D1 : 20 20 20 20 20 20 A0 0D   AD
47D9 : 00 4C 13 35 20 AF 45 EA   D5
47E1 : EA EA AD 2E 24 C9 05 D0   BB
47E9 : 24 20 D6 09 C9 4D 20 D7   2B
47F1 : 45 53 54 45 4E 20 46 4C   35
47F9 : 55 45 53 54 45 52 54 20   C9
4801 : 4A 45 4D 41 4E 44 2E 20   69
4809 : 20 20 20 0D 00 AD 2F 24   56
4811 : C9 05 D0 29 20 D6 09 D6   41
4819 : 4F 4E 20 D7 45 53 54 45   5D
4821 : 4E 20 48 4F 45 52 45 4E   14
4829 : 20 53 49 45 20 D3 43 48   2C
4831 : 52 49 54 54 45 2E 20 20   4E
4839 : 20 20 20 0D 00 AD 2F 24   86
4841 : C9 03 D0 20 20 D6 09 C9   34
4849 : 4D 20 D3 55 45 44 45 4E   6E
4851 : 20 50 4F 4C 54 45 52 54   5B
4859 : 20 45 54 57 41 53 2E 20   C4
4861 : 20 20 0D 00 AD 30 24 C9   55
4869 : 08 D0 25 20 D6 09 CF 42   A0
4871 : 45 4E 20 53 43 48 52 45   9A
4879 : 49 54 20 4A 45 4D 41 4E   9E
4881 : 44 20 C2 45 46 45 48 4C   77
4889 : 45 21 20 20 20 20 0D 00   A2
4891 : 20 D6 09 C1 55 46 20 44   27
4899 : 45 4D 20 C8 4F 46 20 49   E0
48A1 : 53 54 20 45 53 20 52 45   D9
48A9 : 43 48 54 20 4C 41 55 54   F6
48B1 : 21 20 20 20 20 20 0D 0D   40
48B9 : 00 EA EA EA 4C 13 35 EA   4E
48C1 : EA EA 20 19 35 0D FC 09   0D
48C9 : 0D FD 09 0D FF 09 F0 01   C7
48D1 : 60 AD FE 09 C9 04 F0 01   6B
48D9 : 60 AD 21 24 F0 2D 20 D6   83
48E1 : 09 D3 49 45 20 53 43 48   09
48E9 : 4C 49 45 53 53 45 4E 20   6E
48F1 : 44 45 4E 20 D3 43 48 52   8D
48F9 : 41 4E 4B 2E 20 20 20 20   BE
4901 : 0D 0D 00 A9 00 8D 21 24   03
4909 : 4C 13 35 20 D6 09 C4 45   84
4911 : 52 20 D3 43 48 52 41 4E   89
4919 : 4B 20 49 53 54 20 42 45   0B
4921 : 52 45 49 54 53 20 5A 55   3D
4929 : 2E 20 20 20 20 20 0D 0D   C5
4931 : 00 4C 13 35 EA EA EA 20   B5
4939 : 19 35 0D FC 09 0D FD 09   D3
4941 : 0D FF 09 D0 07 AD FE 09   96
4949 : C9 04 F0 01 60 AD 21 24   B1
4951 : D0 2A 20 D6 09 D3 49 45   F8
4959 : 20 4F 45 46 46 4E 45 4E   C4
4961 : 20 44 45 4E 20 D3 43 48   FD
4969 : 52 41 4E 4B 2E 20 20 20   FE
4971 : 20 0D 0D 00 A9 01 8D 21   76
4979 : 24 4C 13 35 20 D6 09 C4   95
4981 : 45 52 20 D3 43 48 52 41   B4
4989 : 4E 4B 20 49 53 54 20 42   BB
4991 : 45 52 45 49 54 53 20 4F   79
4999 : 46 46 45 4E 2E 20 20 20   C2
49A1 : 20 20 0D 0D 00 4C 13 35   CF
49A9 : EA EA EA 20 19 35 0D FD   32
49B1 : 09 0D FF 09 F0 01 60 AD   56
49B9 : FE 09 C9 04 D0 F8 AE FC   B8
49C1 : 09 F0 F3 BD D5 23 C9 01   96
49C9 : D0 45 BD EB 23 C9 03 D0   57
49D1 : 3E 4C 69 4B 20 D6 09 CF   76
49D9 : 4B 2E 20 D3 49 45 20 4C   96
49E1 : 45 47 45 4E 20 45 54 57   11
49E9 : 41 53 20 49 4E 20 44 45   87
49F1 : 4E 20 D3 43 48 52 41 4E   65
49F9 : 4B 2E 20 20 20 20 20 20   2B
4A01 : 0D 00 4C 03 4B EA EA A9   8D
4A09 : 01 9D EB 23 4C 13 35 20   AB
4A11 : D6 09 D3 49 45 20 48 41   83
4A19 : 42 45 4E 20 44 45 4E 20   7D
4A21 : C7 45 47 45 4E 53 54 41   59
4A29 : 4E 44 20 4E 49 43 48 54   E4
4A31 : 21 20 20 20 20 20 [illegible] 0D   C0
4A39 : 00 4C 13 35 EA EA EA 20   BD
4A41 : 19 35 0D FD 09 0D FF 09   03
4A49 : F0 01 60 AD FE 09 C9 04   EF
4A51 : D0 F8 AE FC 09 F0 F3 BD   4C
4A59 : D5 23 C9 04 D0 47 BD EB   C9
4A61 : 23 C9 01 D0 40 4C 3C 4B   B1
4A69 : 20 D6 09 D3 49 45 20 4E   8D
4A71 : 45 48 4D 45 4E 20 45 54   7A
4A79 : 57 41 53 20 41 55 53 20   96
4A81 : 44 45 4D 20 D3 43 48 52   DC
4A89 : 41 4E 4B 2E 20 20 20 20   4E
4A91 : 20 20 20 20 20 20 0D 00   05
4A99 : A9 01 9D D5 23 A9 03 9D   AC
4AA1 : EB 23 4C 13 35 20 D6 09   55
4AA9 : C4 45 52 20 C7 45 47 45   F7
4AB1 : 4E 53 54 41 4E 44 20 49   00
4AB9 : 53 54 20 4E 49 43 48 54   81
4AC1 : 20 49 4D 20 D3 43 48 52   FA
4AC9 : 41 4E 4B 2E 20 20 20 20   8E
4AD1 : 20 20 20 0D 00 4C 13 35   C4
4AD9 : EA EA EA E6 BB E6 BB C6   3F
4AE1 : B7 C6 B7 4C 40 77 4B A5   9F
4AE9 : BA C9 01 D0 08 E6 BB E6   57
4AF1 : BB C6 B7 C6 B7 68 4C D8   78
4AF9 : FF EA EA EA EA EA EA 4C   CF
4B01 : 0E 4B A9 04 AE FC 09 9D   D2
4B09 : D5 23 4C 08 4A C9 08 F0   79
4B11 : 03 4C 02 4E AD FB 09 C9   F7
4B19 : 0C D0 03 4C 1E 41 C9 15   15
4B21 : D0 03 4C 80 4B C9 16 D0   93
4B29 : 03 4C 3D 4C C9 19 D0 03   DA
4B31 : 4C EA 4C C9 1D D0 03 4C   3C
4B39 : 73 4D 60 AD 21 24 C9 01   7D
4B41 : D0 03 4C 69 4A EA EA EA   51
4B49 : 20 D6 09 C4 45 52 20 D3   BE
4B51 : 43 48 52 41 4E 4B 20 49   C7
4B59 : 53 54 20 5A 55 21 20 20   49
4B61 : 20 20 0D 0D 00 4C 13 35   8F
4B69 : AD 21 24 C9 01 D0 D6 4C   74
4B71 : D5 49 AD 21 24 C9 01 D0   B1
4B79 : CC BD D5 23 4C 5B 47 20   FB
4B81 : 19 35 0D FD 09 0D FF 09   43
4B89 : F0 01 60 AD FE 09 C9 02   2B
4B91 : F0 03 4C DB 42 AE FC 09   D1
4B99 : D0 03 4C E9 4B BD D5 23   7B
4BA1 : C9 02 F0 03 4C B5 4B BD   23
4BA9 : EB 23 C9 01 F0 03 4C B5   7C
4BB1 : 4B 4C 6F 47 20 D6 09 C4   4D
4BB9 : 45 52 20 C7 45 47 45 4E   69
4BC1 : 53 54 41 4E 44 20 49 53   6A
4BC9 : 54 20 4E 49 43 48 54 20   F2
4BD1 : 49 4D 20 C2 4F 54 54 49   9D
4BD9 : 43 48 2E 20 20 20 20 20   94
4BE1 : 20 20 20 0D 00 4C 13 35   D4
4BE9 : 20 D6 09 C4 45 52 20 C2   3C
4BF1 : 4F 54 54 49 43 48 20 49   32
4BF9 : 53 54 20 47 52 4F 53 53   FB
4C01 : 2E 20 20 20 20 20 20 20   0F
4C09 : 20 20 20 20 20 20 20 20   09
4C11 : 20 20 20 C4 41 52 49 4E   4A
4C19 : 20 53 45 48 45 4E 20 53   2B
4C21 : 49 45 3A 20 20 20 00 A9   F6
4C29 : 00 8D 09 0B A9 02 8D 3B   EB
4C31 : 32 A9 01 8D 3C 32 20 3E   7C
4C39 : 32 4C 13 35 EA EA EA 20   EF
4C41 : AF 45 AD 2F 24 C9 07 D0   32
4C49 : 2F 20 D6 09 C1 55 53 20   B3
4C51 : 44 45 4D 20 CB 45 4C 4C   40
4C59 : 45 52 20 48 4F 45 52 45   CB
4C61 : 4E 20 53 49 45 20 D3 43   E9
4C69 : 48 52 49 54 54 45 2E 20   20
4C71 : 20 20 20 20 20 20 0D 00   E5
4C79 : AD 30 24 C9 0A D0 25 20   7D
4C81 : D6 09 D6 4F 4E 20 D7 45   4B
4C89 : 53 54 45 4E 20 42 52 55   2A
4C91 : 45 4C 4C 54 20 4A 45 4D   9E
4C99 : 41 4E 44 2E 20 20 20 20   9C
4CA1 : 20 20 0D 00 AD 30 24 C9   95
4CA9 : 0C D0 20 20 D6 09 C9 4E   A3
4CB1 : 20 CE 4F 52 44 45 4E 20   3E
4CB9 : 49 53 54 20 45 53 20 4C   CD
4CC1 : 41 55 54 2E 20 20 20 20   4C
4CC9 : 20 0D 00 20 05 4E 20 D6   65
4CD1 : 09 C5 49 4E 20 C8 55 4E   13
4CD9 : 44 20 42 45 4C 4C 54 2E   3B
4CE1 : 20 20 20 0D 0D 00 4C 13   E3
4CE9 : 35 20 19 35 0D FD 09 0D   1A
4CF1 : FF 09 F0 01 60 AD FE 09   53
4CF9 : C9 02 D0 F8 AE FC 09 F0   EF
4D01 : F3 BD D5 23 C9 01 D0 3D   0F
4D09 : BD EB 23 C9 03 D0 36 20   8E
4D11 : D6 09 D3 49 45 20 4C 45   9B
4D19 : 47 45 4E 20 45 54 57 41   71
4D21 : 53 20 49 4E 20 44 45 4E   76
4D29 : 20 C2 4F 54 54 49 43 48   36
4D31 : 2E 20 20 20 20 0D 0D 00   1A
4D39 : A9 02 9D D5 23 A9 01 9D   C4
4D41 : EB 23 4C 13 35 20 D6 09   F5
4D49 : D3 49 45 20 48 41 42 45   38
4D51 : 4E 20 44 45 4E 20 C7 45   F9
4D59 : 47 45 4E 53 54 41 4E 44   52
4D61 : 20 4E 49 43 48 54 21 20   4F
4D69 : 20 20 20 20 0D 0D 00 4C   77
4D71 : 13 35 20 19 35 0D FD 09   10
4D79 : 0D FF C9 09 F0 01 60 AD FE   AE
4D81 : 09 C9 02 D0 F8 AE FC 09   14
4D89 : F0 F3 BD D5 23 C9 02 D0   C7
4D91 : 3F BD EB 23 C9 01 D0 38   66
4D99 : 20 D6 09 D3 49 45 20 4E   BD
4DA1 : 45 48 4D 45 4E 20 45 54   AA
4DA9 : 57 41 53 20 41 55 53 20   C6
4DB1 : 44 45 4D 20 C2 4F 54 54   90
4DB9 : 49 43 48 2E 20 20 20 20   40
4DC1 : 0D 0D 00 A9 01 9D D5 23   25
4DC9 : A9 03 9D EB 23 4C 13 35   24
4DD1 : 20 D6 09 C4 45 52 20 C7   2E
4DD9 : 45 47 45 4E 53 54 41 4E   56
4DE1 : 44 20 49 53 54 20 4E 49   04
4DE9 : 43 48 54 20 5A 55 20 46   C7
4DF1 : 49 4E 44 45 4E 21 20 20   CA
4DF9 : 20 20 20 20 0D 00 4C 13   5E
4E01 : 35 4C 2C 4E AD 31 24 C9   BA
4E09 : 0C D0 1F 20 D6 09 CE 4F   D9
4E11 : 45 52 44 4C 49 43 48 20   2A
4E19 : 4B 4E 41 4C 4C 54 20 45   D8
4E21 : 54 57 41 53 2E 20 20 20   80
4E29 : 0D 00 60 4C 37 4E A9 00   65
4E31 : 8D 77 24 4C 13 35 C9 09   21
4E39 : F0 03 4C AA 4E AD FB 09   68
4E41 : C9 0C D0 03 4C 1E 41 C9   F3
4E49 : 15 D0 03 4C D8 42 C9 16   04
4E51 : F0 01 60 20 AF 45 AD 2E   16
4E59 : 24 C9 06 D0 2C 20 D6 09   2F
4E61 : D5 4E 54 45 52 20 53 49   21
4E69 : 43 48 20 48 4F 45 52 45   D4
4E71 : 4E 20 53 49 45 20 D3 43   F9
4E79 : 48 52 49 54 54 45 2E 20   30
4E81 : 20 20 20 0D 0D 00 4C 13   83
4E89 : 35 20 D6 09 C5 53 20 49   AF
4E91 : 53 54 20 41 4C 4C 45 53   21
4E99 : 20 52 55 48 49 47 21 20   D4
4EA1 : 20 20 20 0D 0D 00 4C 13   A3
4EA9 : 35 C9 0A F0 03 4C F9 4E   7B
4EB1 : AD FB 09 C9 0C D0 03 4C   C3
4EB9 : 1E 41 C9 15 D0 03 4C D8   95
4EC1 : 42 C9 16 F0 01 60 20 AF   7E
4EC9 : 45 AD 30 24 C9 08 D0 B9   09
4ED1 : 20 D6 09 D6 4F 4D 20 C8   EB
4ED9 : 4F 46 20 48 4F 45 52 54   6E
```

```
4EE1 : 20 4D 41 4E 20 D3 43 48  00
4EE9 : 52 49 54 54 45 21 20 20  9E
4EF1 : 20 20 0D 0D 00 4C 13 35  1F
4EF9 : C9 0B F0 03 4C 4C 4F 20  89
4F01 : A2 4F C9 0C D0 03 4C 1E  D1
4F09 : 41 C9 15 D0 03 4C D8 42  09
4F11 : C9 16 F0 01 60 20 AF 45  92
4F19 : AD 30 24 C9 08 F0 03 4C  CD
4F21 : 8A 4E 20 D6 09 D6 4F 4E  D6
4F29 : 20 D7 45 53 54 45 4E 20  DA
4F31 : 48 4F 45 52 45 4E 20 53  AA
4F39 : 49 45 20 45 54 57 41 53  81
4F41 : 2E 20 20 20 20 0D 0D 00  2A
4F49 : 4C 13 35 C9 0C F0 03 4C  92
4F51 : 9F 4F AD FB 09 C9 0C D0  33
4F59 : 03 4C 1E 41 C9 15 D0 03  C1
4F61 : 4C D8 42 C9 16 F0 01 60  91
4F69 : 20 AF 45 AD 30 24 C9 08  C3
4F71 : F0 03 4C 8A 4E 20 D6 09  9B
4F79 : D6 4F 4E 20 D3 55 45 44  14
4F81 : 45 4E 20 48 4F 45 52 45  F1
4F89 : 4E 20 53 49 45 20 45 54  F8
4F91 : 57 41 53 2E 20 20 20 20  E7
4F99 : 0D 0D 00 4C 13 35 4C AC  1C
4FA1 : 4F 20 24 29 20 00 3E AD  85
4FA9 : FB 09 60 C9 0D F0 03 4C  77
4FB1 : CC 4F 20 A2 4F C9 0C D0  96
4FB9 : 03 4C 1E 41 C9 15 D0 03  21
4FC1 : 4C D8 42 C9 16 F0 01 60  F1
4FC9 : 4C E3 41 4C 30 50 AD D4  C7
4FD1 : 23 F0 01 60 AD FF 09 C9  4B
4FD9 : 08 F0 03 4C F2 4F A2 06  E4
4FE1 : BD 2D 24 CD D3 23 F0 04  1A
4FE9 : CA D0 F5 60 E8 4C 2B 50  E3
4FF1 : EA C9 0F D0 0F AD 39 24  29
4FF9 : CD D3 23 F0 01 60 A9 0E  6D
5001 : 8D FF 09 60 C9 11 D0 0F  63
5009 : AD 3B 24 CD D3 23 F0 01  33
5011 : 60 A9 10 8D FF 09 60 C9  59
5019 : 16 D0 FB AD 40 24 CD D3  50
5021 : 23 F0 01 60 A9 15 8D FF  82
5029 : 09 60 E8 8E FF 09 60 C9  CC
5031 : 0E F0 03 4C 9D 50 AD FB  0D
5039 : 09 C9 0C D0 03 4C 1E 41  D2
5041 : C9 15 D0 03 4C D8 42 C9  51
5049 : 16 F0 01 60 20 D6 09 D3  A8
5051 : 49 45 20 48 4F 45 52 45  41
5059 : 4E 20 4C 41 55 54 45 20  40
5061 : D6 4F 47 45 4C 53 54 49  9D
5069 : 4D 4D 45 4E 2C 20 44 49  DF
5071 : 45 20 20 20 20 20 20 41  D9
5079 : 4C 4C 45 53 20 41 4E 44  75
5081 : 45 52 45 20 55 45 42 45  58
5089 : 52 54 4F 45 4E 45 4E 21  0C
5091 : 20 20 20 20 20 20 0D 0D  1F
5099 : 00 4C 13 35 C9 20 B0 03  91
50A1 : 4C 37 50 C9 20 F0 03 4C  04
50A9 : 8E 52 AD FB 09 C9 0C D0  FC
50B1 : 03 4C 1E 41 C9 0B D0 03  C8
50B9 : 4C D9 50 C9 15 D0 03 4C  BC
50C1 : 17 52 C9 16 D0 03 4C AE  EA
50C9 : 51 C9 17 D0 03 4C 28 51  B5
50D1 : EA EA EA EA EA EA EA 60  BB
50D9 : EA EA EA 20 19 35 0D FC  60
50E1 : 09 0D FD 09 0D FF 09 F0  E8
50E9 : 01 60 AD FE 09 C9 0C D0  16
50F1 : F8 EA EA EA A9 60 8D D3  F2
50F9 : 23 A9 00 8D D4 23 4C B7  AA
5101 : 3D EA EA EA 20 19 35 0D  85
5109 : FC 09 0D FD 09 0D FF 09  98
5111 : F0 01 60 AD FE 09 C9 0C  C7
5119 : D0 F8 EA EA EA A9 20 8D  15
5121 : D3 23 A9 00 4C FC 50 EA  B4
5129 : EA EA 20 19 35 0D FD 09  79
5131 : 0D FD 09 0D FF 09 F0 01  2F
5139 : 60 AD FE 09 C9 03 D0 F8  3A
5141 : AD FC 09 C9 0B D0 F1 AD  42
5149 : E0 23 C9 01 D0 EA AD F6  56
5151 : 23 C9 03 D0 E3 EA EA EA  4B
5159 : 20 D6 09 D3 49 45 20 5A  95
5161 : 45 52 53 54 4F 45 52 45  22
5169 : 4E 20 44 45 4E 20 CB 41  19
5171 : 53 54 45 4E 2E 20 20 20  AE
5179 : 20 20 20 20 20 20 20 20  79
5181 : 20 20 20 C5 54 57 41 53  1E
5189 : 20 4B 4C 49 52 52 54 20  D4
5191 : 4C 45 49 53 45 2E 20 20  C3
5199 : 0D 0D 00 A9 20 8D DF 23  96
51A1 : A9 00 8D F5 23 A9 01 8D  0B
51A9 : 20 24 4C 13 35 EA EA EA  7D
51B1 : 20 AF 45 AD 31 24 C9 1E  47
51B9 : D0 2A 20 D6 09 D6 4F 4E  A2
51C1 : 20 D3 55 45 44 57 45 53  84
51C9 : 54 45 4E 20 4B 4F 4D 4D  56
51D1 : 54 20 4A 45 4D 41 4E 44  11
51D9 : 21 20 20 20 20 20 0D 0D  68
51E1 : 00 4C 13 35 20 D6 09 D3  F7
51E9 : 49 45 20 48 4F 45 52 45  D9
51F1 : 4E 20 44 41 53 20 D7 41  A1
51F9 : 53 53 45 52 20 50 4C 41  CA
5201 : 45 54 53 43 48 45 52 4E  42
5209 : 2E 20 20 20 20 20 20 20  17
5211 : 20 0D 00 4C 13 35 EA EA  9E
5219 : EA 20 19 35 0D FD 09 0D  FF
5221 : FF 09 0D FC 09 F0 01 60  65
5229 : AD FE 09 C9 03 F0 03 4C  2D
5231 : D8 42 AD 20 24 C9 01 F0  10
5239 : 2D 20 D6 09 C4 45 52 20  4D
5241 : CB 41 53 54 45 4E 20 49  E6
5249 : 53 54 20 4E 49 43 48 54  11
5251 : 20 5A 55 20 4F 45 46 46  BC
5259 : 4E 45 4E 21 20 20 20 20  C5
5261 : 0D 0D 00 4C 13 35 20 D6  88
5269 : 09 C4 45 52 20 CB 41 53  7C
5271 : 54 45 4E 20 49 53 54 20  C0
5279 : 5A 45 52 42 52 4F 43 48  90
5281 : 45 4E 2E 20 20 20 20 0D  1B
5289 : 0D 00 4C 13 35 C9 21 D0  D4
5291 : 01 60 C9 22 D0 03 4C 37  3E
5299 : 50 C9 23 F0 F9 C9 60 F0  06
52A1 : 03 4C D8 53 4C ED 52 20  29
52A9 : D6 09 93 11 11 11 11 11  0B
52B1 : 11 11 11 11 11 11 11 C2  15
52B9 : 4F 44 4F 20 CD 45 52 54  FB
52C1 : 49 4E 53 20 50 52 41 45  31
52C9 : 53 45 4E 54 49 45 52 54  8E
52D1 : 3A 0D 00 A9 00 85 FB 85  EE
52D9 : FC A9 04 85 FD C6 FB D0  03
52E1 : FC C6 FC D0 F8 C6 FD D0  F9
52E9 : F4 4C 0F 54 AD FB 09 C9  C4
52F1 : 01 D0 03 4C 77 53 C9 02  E2
52F9 : D0 03 4C 93 53 C9 0C D0  26
5301 : 03 4C 1E 41 C9 15 D0 03  69
5309 : 4C 1B 53 C9 16 D0 03 4C  7D
5311 : 55 53 C9 1F D0 03 4C AF  1C
5319 : 53 60 20 AF 45 EA EA EA  C7
5321 : 20 B7 3D 4C 29 53 60 00  A4
5329 : AD D3 23 8D 27 53 AD D4  A8
5331 : 23 8D 28 53 AD 0D 24 8D  7E
5339 : D3 23 A9 00 8D D4 23 20  54
5341 : 8F 24 EA EA EA AD 27 53  5A
5349 : 8D D3 23 AD 28 53 8D D4  3B
5351 : 23 4C 13 35 20 AF 45 20  DB
5359 : D6 09 D7 41 53 53 45 52  5B
5361 : 20 50 4C 41 45 54 53 43  AF
5369 : 48 45 52 54 21 20 20 20  47
5371 : 0D 0D 00 4C 13 35 20 AF  49
5379 : 45 4C E4 53 C9 20 F0 0D  03
5381 : C9 21 F0 01 60 A9 24 20  5B
5389 : DB 53 4C 1E 53 A9 21 4C  84
5391 : 88 53 20 AF 45 4C F8 53  02
5399 : C9 21 F0 0D C9 24 F0 01  54
53A1 : 60 A9 21 4C 88 53 4C 88  0D
53A9 : 53 A9 20 4C 88 53 EA EA  07
53B1 : EA 20 19 35 0D FC 09 0D  8F
53B9 : FD 09 0D FF 09 F0 01 60  5B
53C1 : AD FE 09 C9 0C D0 F8 EA  6A
53C9 : EA EA AD 0D 24 C9 21 F0  2C
53D1 : EE AD 0D 24 4C 20 51 4C  01
53D9 : 0C 54 8D 0D 24 A9 00 8D  BF
53E1 : FB 09 60 AD DF 23 C9 01  6F
53E9 : D0 F8 AD F5 23 C9 03 D0  8E
53F1 : F1 AD 0D 24 4C 7D 53 AD  DA
53F9 : DF 23 C9 01 D0 E4 AD F5  D3
5401 : 23 C9 03 D0 DD AD 0D 24  AB
5409 : 4C 99 53 4C 16 54 A9 00  2B
5411 : 85 C6 4C 99 3F C9 24 F0  F4
5419 : 03 4C 09 58 AD FB 09 C9  02
5421 : 0B D0 03 4C D9 50 C9 0C  3E
5429 : D0 03 4C 1E 41 C9 15 D0  AA
5431 : 03 4C 59 54 C9 16 D0 03  D2
5439 : 4C 85 55 C9 18 D0 03 4C  83
5441 : E0 55 C9 19 D0 03 4C D3  5F
5449 : 56 C9 1C D0 03 4C 5A 56  4E
5451 : C9 1D D0 03 4C 71 57 60  AC
5459 : 20 19 35 0D FD 09 0D FF  51
5461 : 09 F0 01 60 AD FE 09 C9  B9
5469 : 08 F0 03 4C D8 42 AE FC  88
5471 : 09 D0 03 4C CA 54 AD 25  7D
5479 : 24 C9 01 F0 03 4C 59 55  83
5481 : BD D5 23 C9 08 F0 03 4C  D8
5489 : 98 54 BD EB 23 C9 01 F0  9E
5491 : 03 4C 98 54 4C 6F 47 20  09
5499 : D6 09 C4 45 52 20 C7 45  9D
54A1 : 47 45 4E 53 54 41 4E 44  9A
54A9 : 20 49 53 54 20 4E 49 43  ED
54B1 : 48 54 20 49 4D 20 C3 4F  D8
54B9 : 4E 54 41 49 4E 45 52 2E  60
54C1 : 20 20 20 20 0D 00 4C 13  26
54C9 : 35 AD 25 24 C9 01 F0 35  76
54D1 : 20 D6 09 D3 49 45 20 53  FF
54D9 : 45 48 45 4E 20 45 49 4E  4B
54E1 : 45 4E 20 47 45 53 43 48  CB
54E9 : 4C 4F 53 53 45 4E 45 4E  95
54F1 : 20 C3 4F 4E 54 41 49 4E  A2
54F9 : 45 52 2E 20 20 20 20 20  BB
5501 : 0D 00 4C 13 35 20 D6 09  46
5509 : D3 49 45 20 53 45 48 45  E1
5511 : 4E 20 45 49 4E 45 4E 20  72
5519 : 4F 46 46 45 4E 45 4E 20  4E
5521 : C3 4F 4E 54 41 49 4E 45  CC
5529 : 52 2E 20 20 20 20 20 20  62
5531 : C4 41 52 49 4E 20 53 45  11
5539 : 48 45 4E 20 53 49 45 3A  C5
5541 : 20 20 00 A9 00 8D 09 0B  4D
5549 : A9 08 8D 3B 32 A9 01 8D  51
5551 : 3C 32 20 3E 32 4C 13 35  B2
5559 : 20 D6 09 C4 45 52 20 C3  AE
5561 : 4F 4E 54 41 49 4E 45 52  D5
5569 : 20 49 53 54 20 47 45 53  85
5571 : 43 48 4C 4F 53 53 45 4E  57
5579 : 21 20 20 20 20 20 0D 0D  08
5581 : 00 4C 13 35 20 AF 45 A2  ED
5589 : 10 BD 2B 24 C9 25 F0 03  57
5591 : 4C C0 55 20 D6 09 C9 4D  0E
5599 : 20 CE 4F 52 44 45 4E 20  26
55A1 : 4C 41 45 55 46 54 20 45  9C
55A9 : 49 4E 45 20 D3 54 52 45  22
55B1 : 49 46 45 21 20 20 20 20  57
55B9 : A0 0D 0D 00 4C 13 35 20  96
55C1 : D6 09 C5 49 4E 45 20 CD  E2
55C9 : 4F 45 57 45 20 53 43 48  74
55D1 : 52 45 49 54 2E 20 20 20  47
55D9 : 20 0D 0D 00 4C 13 35 20  36
55E1 : 19 35 0D FC 09 0D FD 09  7B
55E9 : 0D FF 09 F0 01 60 AD FE  1E
55F1 : 09 C9 08 D0 F8 EA EA EA  63
55F9 : A2 08 BD 1D 24 C9 00 F0  25
5601 : 2E A9 00 9D 1D 24 20 D6  D9
5609 : 09 D3 49 45 20 53 43 48  31
5611 : 4C 49 45 53 53 45 4E 20  96
5619 : 44 45 4E 20 C3 4F 4E 54  30
5621 : 41 49 4E 45 52 2E 20 20  9A
5629 : 20 0D 0D 00 4C 13 35 20  86
5631 : D6 09 C4 45 52 20 C3 4F  39
5639 : 4E 54 41 49 4E 45 52 20  C3
5641 : 57 41 52 20 42 45 52 45  F4
5649 : 49 54 53 20 5A 55 2E 20  DF
5651 : 20 20 20 0D 0D 00 4C 13  53
5659 : 35 20 19 35 0D FC 09 0D  82
5661 : FD 09 0D FF 09 F0 01 60  03
5669 : AD FE 09 C9 08 D0 F8 EA  D2
5671 : A2 08 BD 1D 24 C9 01 F0  A1
5679 : 2D A9 01 9D 1D 24 20 D6  90
5681 : 09 D3 49 45 20 4F 45 46  8D
5689 : 46 4E 45 4E 20 44 45 4E  E7
5691 : 20 C3 4F 4E 54 41 49 4E  42
5699 : 45 52 2E 20 20 20 20 20  5B
56A1 : 0D 0D 00 4C 13 35 20 D6  C8
56A9 : 09 C4 45 52 20 C3 4F 4E  AA
56B1 : 54 41 49 4E 45 52 20 57  D8
56B9 : 41 52 20 42 45 52 45 49  02
56C1 : 54 53 20 4F 46 46 45 4E  F9
56C9 : 2E 20 20 20 0D 0D 00 4C  E5
56D1 : 13 35 EA EA EA 20 19 35  15
56D9 : 0D FD 09 0D FF 09 F0 01  D7
56E1 : 60 AD FE 09 C9 08 D0 F8  0B
56E9 : AE FC 09 F0 F3 BD D5 23  40
56F1 : C9 01 D0 4D BD EB 23 C9  74
56F9 : 03 D0 46 AD 25 24 C9 01  48
5701 : F0 03 4C 45 57 20 D6 09  12
5709 : CF 4B 2E 20 D3 49 45 20  EA
5711 : 4C 45 47 45 4E 20 45 54  1E
5719 : 57 41 53 20 49 4E 20 44  FA
5721 : 45 4E 20 C3 4F 4E 54 41  49
5729 : 49 4E 45 52 2E 20 20 20  DA
5731 : 20 0D 0D 00 A9 08 9D D5  18
5739 : 23 A9 01 9D EB 23 4C 13  54
5741 : 35 4C 46 4D 20 D6 09 C4  3E
5749 : 45 52 20 C3 4F 4E 54 41  73
5751 : 49 4E 45 52 20 49 53 54  9F
5759 : 20 47 45 53 43 48 4C 4F  1F
5761 : 53 53 45 4E 21 20 20 20  4D
5769 : 20 20 0D 0D 00 4C 13 35  97
5771 : 20 19 35 0D FD 09 0D FF  69
5779 : 09 F0 01 60 AD FE 09 C9  D1
5781 : 08 D0 F8 AE FC 09 F0 F3  C9
5789 : BD D5 23 C9 08 D0 4C BD  E7
5791 : EB 23 C9 01 D0 45 AD 25  D9
5799 : 24 C9 01 F0 03 4C DF 57  C1
57A1 : 20 D6 09 D3 49 45 20 4E  C5
57A9 : 45 48 4D 45 4E 20 45 54  B2
57B1 : 57 41 53 20 41 55 53 20  CE
57B9 : 44 45 4D 20 C3 4F 4E 54  90
57C1 : 41 49 4E 45 52 2E 20 20  3A
57C9 : 20 20 20 20 0D 00 A9 01  7F
57D1 : 9D D5 23 A9 03 9D EB 23  6A
57D9 : 4C 13 35 4C D1 4D 20 D6  3B
57E1 : 09 C4 45 52 20 C3 4F 4E  E2
57E9 : 54 41 49 4E 45 52 20 49  F4
57F1 : 53 54 20 47 45 53 43 48  EC
57F9 : 4C 4F 53 53 45 4E 21 20  B8
5801 : 20 20 0D 0D 00 4C 13 35  2F
5809 : C9 25 F0 03 4C 4A 59 AD  D9
5811 : FB 09 C9 0C F0 03 4C 08  ED
5819 : 59 20 AF 45 20 D6 09 D3  9B
5821 : 49 45 20 48 41 42 45 20  99
```

**Listing »Freiheit« (Fortsetzung)**

```
5829 : 4E 4F 43 48 20 49 48 52  0B
5831 : 45 4E 20 D3 54 52 41 45  87
5839 : 46 4C 49 4E 47 53 41 4E  72
5841 : 5A 55 47 20 20 20 20 41  22
5849 : 4E 2E 20 C1 45 4E 44 45  51
5851 : 52 4E 20 53 49 45 20 44  05
5859 : 41 53 2C 20 53 4F 4E 53  E3
5861 : 54 20 57 45 52 44 45 4E  3D
5869 : 20 53 49 45 0D 56 4F 4E  8B
5871 : 20 4A 45 44 45 52 20 D3  9F
5879 : 54 52 45 49 46 45 20 45  0A
5881 : 52 4B 41 4E 4E 54 21 0D  B9
5889 : C1 55 53 53 45 52 48 41  BF
5891 : 4C 42 20 44 45 53 20 CF  9E
5899 : 52 54 45 53 20 47 49 42  B7
58A1 : 54 20 45 53 20 49 52 47  E5
58A9 : 45 4E 44 57 4F 20 20 20  CB
58B1 : 45 49 4E 45 20 D0 4F 4C  35
58B9 : 49 5A 45 49 4B 4F 4E 54  BB
58C1 : 52 4F 4C 4C 45 2C 20 57  3C
58C9 : 4F 20 53 49 45 20 49 4E  3D
58D1 : 20 4A 45 44 45 4D 0D C6  71
58D9 : 41 4C 4C 45 20 45 52 4B  08
58E1 : 41 4E 4E 54 20 57 55 45  04
58E9 : 52 44 45 4E 21 20 20 20  4C
58F1 : 20 20 20 20 20 20 20 20  F1
58F9 : 20 20 20 20 20 20 20 20  F9
5901 : 20 20 0D 00 4C 13 35 C9  3A
5909 : 15 D0 03 4C D8 42 C9 16  C4
5911 : D0 36 20 AF 45 20 D6 09  BD
5919 : D3 49 45 20 48 4F 45 52  9F
5921 : 45 4E 20 4E 55 52 20 4C  60
5929 : 41 55 54 45 20 C1 4C 4C  AC
5931 : 54 41 47 53 47 45 52 41  CD
5939 : 45 55 53 43 48 45 2E 20  0E
5941 : 20 20 20 0D 00 4C 13 35  34
5949 : 60 C9 26 F0 03 4C 8D 59  B1
5951 : AD FB 09 C9 0C D0 03 4C  63
5959 : 1E 41 C9 15 D0 03 4C D8  35
5961 : 42 C9 16 F0 01 60 20 AF  1E
5969 : 45 20 D6 09 D3 49 45 20  72
5971 : 48 4F 45 52 45 4E 20 4C  DC
5979 : 45 49 53 45 20 CD 55 53  4D
5981 : 49 4B 2E 20 20 20 0D 0D  51
5989 : 00 4C 13 35 C9 27 F0 03  BA
5991 : 4C AD 59 AD FB 09 C9 0C  07
5999 : D0 03 4C 1E 41 C9 15 D0  1A
59A1 : 03 4C D8 42 C9 16 F0 01  5C
59A9 : 60 4C 13 59 C9 28 F0 03  C7
59B1 : 4C F7 59 AD FB 09 C9 0C  4C
59B9 : D0 03 4C 1E 41 C9 15 D0  3A
59C1 : 03 4C D8 42 C9 16 F0 01  7C
59C9 : 60 20 AF 45 20 D6 09 D3  52
59D1 : 49 45 20 48 4F 45 52 45  C1
59D9 : 4E 20 4E 49 43 48 54 53  63
59E1 : 20 C2 45 53 4F 4E 44 45  21
59E9 : 52 45 53 2E 20 20 20 20  3C
59F1 : 0D 0D 00 4C 13 35 C9 29  63
59F9 : D0 03 4C 37 50 C9 2A F0  23
5A01 : 03 4C B7 5B AD FB 09 C9  F6
5A09 : 0C D0 03 4C 1E 41 C9 15  05
5A11 : D0 03 4C 2C 5A C9 16 D0  E9
5A19 : 03 4C 13 59 C9 19 D0 03  E1
5A21 : 4C F4 5A C9 1D D0 03 4C  B4
5A29 : 53 5B 60 20 19 35 0D FD  B1
5A31 : 09 0D FF 09 F0 01 60 AD  D6
5A39 : FE 09 C9 06 F0 03 4C D8  F9
5A41 : 42 AE FC 09 D0 03 4C 9F  D0
5A49 : 5A BD D5 23 C9 06 F0 03  F2
5A51 : 4C 61 5A BD EB 23 C9 01  9D
5A59 : F0 03 4C 61 5A 4C 6F 47  5E
5A61 : 20 D6 09 C4 45 52 20 C7  BE
5A69 : 45 47 45 4E 53 54 41 4E  E6
5A71 : 44 20 49 53 54 20 4E 49  94
5A79 : 43 48 54 20 49 4E 20 44  0A
5A81 : 45 4D 20 20 20 20 20 20  3D
5A89 : 20 20 20 C2 45 48 41 45  40
5A91 : 4C 54 45 52 2E 20 20 20  48
5A99 : 0D 0D 00 4C 13 35 20 D6  C0
5AA1 : 09 D3 49 45 20 53 45 48  D1
5AA9 : 45 4E 20 45 49 4E 45 4E  7F
5AB1 : 20 C2 45 48 41 45 4C 54  A5
5AB9 : 45 52 2E 20 20 20 20 20  7B
5AC1 : 20 20 20 20 20 20 20 20  C1
5AC9 : 20 C4 41 52 49 4E 20 53  14
5AD1 : 45 48 45 4E 20 53 49 45  A2
5AD9 : 3A 20 20 20 20 00 A9 00  D8
5AE1 : 8D 09 0B A9 06 8D 3B 32  09
5AE9 : A9 01 8D 3C 32 20 3E 32  7F
5AF1 : 4C 13 35 20 19 35 0D FD  84
5AF9 : 09 0D FF 09 F0 01 60 AD  9E
5B01 : FE 09 C9 06 D0 F8 AE FC  40
5B09 : 09 F0 F3 BD D5 23 C9 01  DE
5B11 : D0 3D BD EB 23 C9 03 D0  9B
5B19 : 36 20 D6 09 D3 49 45 20  13
5B21 : 4C 45 47 45 4E 20 45 54  2E
5B29 : 57 41 53 20 49 4E 20 44  0A
5B31 : 45 4E 20 C2 45 48 41 45  24
5B39 : 4C 54 45 52 2E 20 20 0D  C9
5B41 : 0D 00 A9 06 9D D5 23 A9  E2
5B49 : 01 9D EB 23 4C 13 35 4C  43
5B51 : 46 4D 20 19 35 0D FD 09  2F
5B59 : 0D FF 09 F0 01 60 AD FE  8E
5B61 : 09 C9 06 D0 F8 AE FC 09  F5
5B69 : F0 F3 BD D5 23 C9 06 D0  B7
5B71 : 42 BD EB 23 C9 01 D0 3B  4F
5B79 : 20 D6 09 D3 49 45 20 4E  9D
5B81 : 45 48 4D 45 4E 20 45 54  8A
5B89 : 57 41 53 20 41 55 53 20  A6
5B91 : 44 45 4D 20 C2 45 48 41  C9
5B99 : 45 4C 54 45 52 2E 20 20  19
5BA1 : 20 20 20 20 0D 00 A9 01  57
5BA9 : 9D D5 23 A9 03 9D EB 23  42
5BB1 : 4C 13 35 4C D1 4D C9 2B  63
5BB9 : F0 03 4C D7 5B AD FB 09  5E
5BC1 : C9 0C D0 03 4C 1E 41 C9  73
5BC9 : 15 D0 03 4C D8 42 C9 16  84
5BD1 : F0 01 60 4C 13 59 C9 2C  5F
5BD9 : F0 03 4C 7E 5D AD FB 09  73
5BE1 : C9 0C D0 03 4C 1E 41 C9  93
5BE9 : 15 D0 03 4C 05 5C C9 16  37
5BF1 : D0 03 4C 13 59 C9 19 D0  A2
5BF9 : 03 4C C0 5C C9 1D D0 03  AD
5C01 : 4C 20 5D 60 20 13 35 0D  4A
5C09 : FD 09 0D FF 09 F0 01 60  AB
5C11 : AD FE 09 C9 0B F0 03 4C  96
5C19 : D8 42 AE FC 09 D0 03 4C  19
5C21 : 6D 5C BD D5 23 C9 0B F0  75
5C29 : 03 4C 3A 5C BD EB 23 C9  C8
5C31 : 01 F0 03 4C 3A 5C 4C 6F  8B
5C39 : 47 20 D6 09 C4 45 52 20  67
5C41 : C7 45 47 45 4E 53 54 41  79
5C49 : 4E 44 20 49 53 54 20 4E  E0
5C51 : 49 43 48 54 20 49 4E 20  9E
5C59 : 44 45 52 20 D4 4F 4E 4E  76
5C61 : 45 2E 20 20 20 20 20 0D  67
5C69 : 00 4C 13 35 20 D6 09 D3  7F
5C71 : 49 45 20 53 45 48 45 4E  18
5C79 : 20 45 49 4E 45 20 D4 4F  9F
5C81 : 4E 4E 45 2E 20 20 20 20  D1
5C89 : 20 20 20 20 20 20 20 20  89
5C91 : 20 20 20 20 20 20 20 C4  DB
5C99 : 41 52 49 4E 20 53 45 48  62
5CA1 : 45 4E 20 53 49 45 3A 20  68
5CA9 : 20 00 A9 00 8D 09 0B A9  D4
5CB1 : 0B 8D 3B 32 A9 01 8D 3C  E9
5CB9 : 32 20 3E 32 4C 13 35 20  44
5CC1 : 19 35 0D FD 09 0D FF 09  83
5CC9 : F0 01 60 AD FE 09 C9 0B  7D
5CD1 : D0 F8 AE FC 09 F0 F3 BD  CC
5CD9 : D5 23 C9 01 D0 3E BD EB  A0
5CE1 : 23 C9 03 D0 37 20 D6 09  A6
5CE9 : CF 4B 2E 20 D3 49 45 20  CA
5CF1 : 4C 45 47 45 4E 20 45 54  FE
5CF9 : 57 41 53 20 49 4E 20 44  DA
5D01 : 49 45 20 D4 4F 4E 4E 45  BB
5D09 : 2E 20 20 20 0D 0D 00 A9  E0
5D11 : 0B 9D D5 23 A9 01 9D EB  B6
5D19 : 23 4C 13 35 4C 46 4D 20  3A
5D21 : 19 35 0D FD 09 0D FF 09  E3
5D29 : F0 01 60 AD FE 09 C9 0B  DD
5D31 : D0 F8 AE FC 09 F0 F3 BD  2C
5D39 : D5 23 C9 0B D0 3C BD EB  31
5D41 : 23 C9 01 D0 35 20 D6 09  65
5D49 : D3 49 45 20 4E 45 48 4D  E1
5D51 : 45 4E 20 45 54 57 41 53  1A
5D59 : 20 41 55 53 20 44 45 52  B7
5D61 : 20 D4 4F 4E 4E 45 2E 20  91
5D69 : 20 20 0D 0D 00 A9 01 9D  0B
5D71 : D5 23 A9 03 9D EB 23 4C  01
5D79 : 13 35 4C D1 4D C9 2D F0  2E
5D81 : 03 4C BF 5D AD FB 09 C9  B8
5D89 : 0C D0 03 4C 1E 41 C9 15  85
5D91 : D0 03 4C D8 42 C9 16 F0  BE
5D99 : 01 60 20 AF 45 20 D6 09  8B
5DA1 : C5 53 20 49 53 54 20 41  1C
5DA9 : 4C 4C 45 53 20 52 55 48  52
5DB1 : 49 47 21 20 20 20 20 20  AE
5DB9 : 0D 0D 00 4C 13 35 C9 2E  35
5DC1 : D0 03 4C 85 5D C9 2F F0  99
5DC9 : F9 C9 30 F0 03 4C EA 5D  CA
5DD1 : AD FB 09 C9 0C D0 03 4C  E3
5DD9 : 1E 41 C9 15 D0 03 4C D8  B5
5DE1 : 42 C9 16 F0 01 60 4C 13  16
5DE9 : 59 C9 31 F0 03 4C 69 5E  86
5DF1 : AD FB 09 C9 0C D0 03 4C  03
5DF9 : 1E 41 C9 15 D0 03 4C 2B  79
5E01 : 5E C9 16 D0 24 20 AF 45  70
5E09 : 20 D6 09 C5 49 4E 49 47  4A
5E11 : 45 20 D5 48 52 45 4E 20  AE
5E19 : 54 49 43 4B 45 4E 2E 20  0C
5E21 : 20 20 20 0D 0D 00 4C 13  23
5E29 : 35 60 20 19 35 0D FC 09  7B
5E31 : 0D FD 09 0D FF 09 F0 01  2F
5E39 : 60 AD FE 09 C9 09 F0 03  FF
5E41 : 4C D8 42 20 D6 09 C4 49  E9
5E49 : 45 20 C1 4C 41 52 4D 41  F7
5E51 : 4E 4C 41 47 45 20 49 53  20
5E59 : 54 20 41 4E 21 21 21 20  B7
5E61 : 20 20 0D 0D 00 4C 13 35  8F
5E69 : C9 32 F0 03 4C 89 5E AD  CE
5E71 : FB 09 C9 0C D0 03 4C 1E  77
5E79 : 41 C9 15 D0 03 4C D8 42  79
5E81 : C9 16 F0 01 60 4C 13 59  19
5E89 : C9 33 F0 03 4C F4 61 AD  D6
5E91 : FB 09 C9 0C D0 03 4C 1E  97
5E99 : 41 C9 15 D0 03 4C C5 5E  85
5EA1 : C9 16 D0 03 4C 13 59 C9  60
5EA9 : 18 D0 03 4C C9 5F C9 19  65
5EB1 : D0 03 4C B6 60 C9 1C D0  53
5EB9 : 03 4C 3E 60 C9 1D D0 03  4D
5EC1 : 4C 51 61 60 20 19 35 0D  D4
5EC9 : FD 09 0D FF 09 F0 01 60  6B
5ED1 : AD FE 09 C9 0A F0 03 4C  46
5ED9 : D8 42 AE FC 09 D0 03 4C  D9
5EE1 : 43 5F AD 27 24 C9 01 F0  9B
5EE9 : 03 4C 4A 5F BD D5 23 C9  3B
5EF1 : 0A F0 03 4C 04 5F BD EB  C8
5EF9 : 23 C9 01 F0 03 4C 04 5F  C1
5F01 : 4C 6F 47 20 D6 09 C4 45  2E
5F09 : 52 20 C7 45 47 45 4E 53  84
5F11 : 54 41 4E 44 20 49 53 54  64
5F19 : 20 4E 49 43 48 54 20 49  55
5F21 : 4E 20 44 45 52 20 20 20  20
5F29 : 20 20 20 20 20 20 CD 55  4A
5F31 : 45 4C 4C 54 4F 4E 4E 45  65
5F39 : 2E 20 20 20 0D 0D 00 4C  55
5F41 : 13 35 AD 27 24 C9 01 F0  B6
5F49 : 2B 20 D6 09 C4 49 45 20  47
5F51 : CD 55 45 4C 4C 54 4F 4E  E5
5F59 : 4E 45 20 49 53 54 20 47  62
5F61 : 45 53 43 48 4C 4F 53 53  5D
5F69 : 45 4E 2E 20 20 20 0D 0D  B6
5F71 : 00 4C 13 35 20 D6 09 C4  69
5F79 : 49 45 20 CD 55 45 4C 4C  70
5F81 : 54 4F 4E 4E 45 20 49 53  FB
5F89 : 54 20 4F 46 46 45 4E 2E  AE
5F91 : 20 20 20 20 20 20 20 20  91
5F99 : 20 20 20 20 20 20 20 C4  E3
5FA1 : 41 52 49 4E 20 53 45 48  6A
5FA9 : 45 4E 20 53 49 45 3A 20  70
5FB1 : 20 20 00 A9 00 8D 09 0B  BD
5FB9 : A9 0A 8D 3B 32 A9 01 8D  C2
5FC1 : 3C 32 20 3E 32 4C 13 35  22
5FC9 : 20 19 35 0D FC 09 0D FD  AD
5FD1 : 09 0D FF 09 F0 01 60 AE  78
5FD9 : FE 09 E0 0A D0 F8 BD 1D  DB
5FE1 : 24 C9 00 F0 2E 20 D6 09  59
5FE9 : D3 49 45 20 53 43 48 4C  BF
5FF1 : 49 45 53 53 45 4E 20 44  EC
5FF9 : 49 45 20 CD 55 45 4C 4C  F0
6001 : 54 4F 4E 4E 45 2E 20 20  E1
6009 : 0D 0D 00 A9 00 9D 1D 24  7C
6011 : 4C 13 35 20 D6 09 C4 49  94
6019 : 45 20 CD 55 45 4C 4C 54  1D
6021 : 4F 4E 4E 45 20 57 41 52  3A
6029 : 20 42 45 52 45 49 54 53  9D
6031 : 20 5A 55 2E 20 20 20 0D  37
6039 : 0D 00 4C 13 35 20 19 35  DF
6041 : 0D FC 09 0D FD 09 0D FF  0D
6049 : 09 F0 01 60 AE FE 09 E0  DF
6051 : 0A D0 F8 BD 1D 24 C9 01  D5
6059 : F0 2D 20 D6 09 D3 49 45  A2
6061 : 20 4F 45 46 46 4E 45 4E  CC
6069 : 20 44 49 45 20 CD 55 45  F7
6071 : 4C 4C 54 4F 4E 4E 45 2E  AB
6079 : 20 20 20 20 0D 0D 00 A9  42
6081 : 01 9D 1D 24 4C 13 35 20  8F
6089 : D6 09 C4 49 45 20 CD 55  75
6091 : 45 4C 4C 54 4F 4E 4E 45  C5
6099 : 20 57 41 52 20 42 45 52  CD
60A1 : 45 49 54 53 20 4F 46 46  2C
60A9 : 45 4E 2E 20 20 20 20 0D  43
60B1 : 0D 00 4C 13 35 20 19 35  57
60B9 : 0D FD 09 0D FF 09 F0 01  B7
60C1 : 60 AD FE 09 C9 0A D0 F8  FB
60C9 : AE FC 09 F0 F3 BD D5 23  20
60D1 : C9 01 D0 4D BD EB 23 C9  54
60D9 : 03 D0 46 AD 27 24 C9 01  48
60E1 : F0 03 4C 25 61 20 D6 09  8F
60E9 : CF 4B 2E 20 D3 49 45 20  CA
60F1 : 4C 45 47 45 4E 20 45 54  FE
60F9 : 57 41 53 20 49 4E 20 44  DA
6101 : 49 45 20 CD 55 45 4C 4C  F8
6109 : 54 4F 4E 4E 45 2E 20 20  E9
6111 : 20 20 0D 00 A9 0A 9D D5  92
6119 : 23 A9 01 9D EB 23 4C 13  34
6121 : 35 4C 46 4D 20 D6 09 C4  1E
6129 : 49 45 20 CD 55 45 4C 4C  20
6131 : 54 4F 4E 4E 45 20 49 53  AB
6139 : 54 20 47 45 53 43 48 4C  21
6141 : 4F 53 53 45 4E 21 20 20  66
6149 : 20 20 0D 0D 00 4C 13 35  77
6151 : 20 19 35 0D FD 09 0D FF  49
6159 : 09 F0 01 60 AD FE 09 C9  B1
```

```
6161 : 0A D0 F8 AE FC 09 F0 F3  AB
6169 : BD D5 23 C9 0A D0 55 BD  0B
6171 : EB 23 C9 01 D0 4E AD 27  05
6179 : 24 C9 01 F0 03 4C C8 61  59
6181 : 20 D6 09 D3 49 45 20 4E  A5
6189 : 45 48 4D 45 4E 20 45 54  92
6191 : 57 41 53 20 41 55 53 20  AE
6199 : 44 45 52 20 CD 55 45 4C  4E
61A1 : 4C 54 4F 4E 4E 45 2E 20  BD
61A9 : 20 20 20 D0 46 55 49 21  72
61B1 : 20 20 20 20 0D 0D 00 A9  7A
61B9 : 01 9D D5 23 A9 03 9D EB  64
61C1 : 23 4C 13 35 4C D1 4D 20  3F
61C9 : D6 09 C4 49 45 20 CD 55  B5
61D1 : 45 4C 4C 54 4F 4E 4E 45  05
61D9 : 20 49 53 54 20 47 45 53  F5
61E1 : 43 48 4C 4F 53 53 45 4E  C7
61E9 : 2E 20 20 20 20 0D 0D 00  D2
61F1 : 4C 13 35 C9 34 D0 03 4C  BC
61F9 : 70 5E C9 25 D0 03 4C 37  74
6201 : 50 C9 3E 90 F9 C9 40 F0  A8
6209 : 03 4C C6 63 AD FB 09 C9  C3
6211 : 16 D0 22 20 AF 45 20 D6  6F
6219 : 09 C5 49 4E 20 D5 48 55  9D
6221 : 20 4B 52 41 45 43 48 54  DC
6229 : 53 54 21 20 20 20 20 0D  90
6231 : 0D 00 4C 13 35 C9 15 D0  4B
6239 : 03 4C 53 62 C9 0C D0 03  CA
6241 : 4C 1E 41 C9 19 D0 03 4C  E3
6249 : 0A 63 C9 1D D0 03 4C 69  44
6251 : 63 60 20 19 35 0D FD 09  D5
6259 : 0D FF 09 F0 01 60 AD FE  8E
6261 : 09 C9 07 F0 03 4C D8 42  A9
6269 : AE FC 09 D0 03 4C B6 62  24
6271 : BD D5 23 C9 07 F0 03 4C  B7
6279 : 88 62 BD EB 23 C9 01 F0  85
6281 : 03 4C 88 62 4C 6F 47 20  B6
6289 : D6 09 C4 45 52 20 C7 45  8D
6291 : 47 45 4E 53 54 41 4E 44  8A
6299 : 20 49 53 54 20 4E 49 43  DD
62A1 : 48 54 20 49 4D 20 D2 4F  04
62A9 : 48 52 2E 20 20 20 20 0D  48
62B1 : 0D 00 4C 13 35 20 D6 09  F6
62B9 : D3 49 45 20 53 45 48 45  91
62C1 : 4E 20 45 49 4E 20 4C 41  33
62C9 : 4E 47 45 53 20 D2 4F 48  DD
62D1 : 52 2E 20 20 20 20 20 20  0A
62D9 : 20 20 20 20 20 20 20 20  D9
62E1 : C4 41 52 49 4E 20 53 45  C1
62E9 : 48 45 4E 20 53 49 45 3A  75
62F1 : 20 20 20 00 A9 00 8D 09  0C
62F9 : 0B A9 07 8D 3B 32 A9 01  3A
6301 : 8D 3C 32 20 3E 32 4C 13  0A
6309 : 35 20 19 35 0D FD 09 0D  3A
6311 : FF 09 F0 01 60 AD FE 09  73
6319 : C9 07 D0 F8 AE FC 09 F0  92
6321 : F3 BD D5 23 C9 01 D0 3D  2F
6329 : BD EB 23 C9 03 D0 36 20  AE
6331 : D6 09 CF 4B 2E 20 D3 49  AF
6339 : 45 20 4C 45 47 45 4E 20  62
6341 : 45 54 57 41 53 20 49 4E  A6
6349 : 20 44 41 53 20 D2 4F 48  AC
6351 : 52 2E 20 20 20 0D 0D 00  65
6359 : A9 07 9D D5 23 A9 01 9D  67
6361 : EB 23 4C 13 35 4C 46 4D  BD
6369 : 20 19 35 0D FD 09 0D FF  61
6371 : 09 F0 01 60 AD FE 09 C9  C9
6379 : 07 D0 F8 AE FC 09 F0 F3  C0
6381 : BD D5 23 C9 07 D0 3B BD  8A
6389 : EB 23 C9 01 D0 34 20 D6  75
6391 : 09 D3 49 45 20 4E 45 48  99
6399 : 4D 45 4E 20 45 54 57 41  F7
63A1 : 53 20 41 55 53 20 44 45  D1
63A9 : 4D 20 D2 4F 48 52 2E 20  B5
63B1 : 20 20 0D 0D 00 A9 01 9D  53
63B9 : D5 23 A9 03 9D EB 23 4C  49
63C1 : 13 35 4C D1 4D C9 41 D0  86
63C9 : 03 4C 37 50 C9 42 F0 03  43
63D1 : 4C 86 67 AD FB 09 C9 0C  37
63D9 : D0 03 4C 1E 41 C9 15 D0  5A
63E1 : 03 4C 09 64 C9 16 D0 03  70
63E9 : 4C 13 59 C9 18 D0 03 4C  FB
63F1 : 47 66 C9 19 D0 03 4C 26  A3
63F9 : 65 C9 1C D0 03 4C 02 67  CD
6401 : C9 1D D0 03 4C B7 65 60  C6
6409 : 20 19 35 0D FD 09 0D FF  01
6411 : 09 F0 01 60 AD FE 09 C9  69
6419 : 01 F0 03 4C D8 42 AE FC  31
6421 : 09 D0 03 4C 79 64 AD 1E  8A
6429 : 24 C9 01 F0 03 4C FE 64  E8
6431 : BD D5 23 C9 01 F0 03 4C  17
6439 : 48 64 BD EB 23 C9 01 F0  06
6441 : 03 4C 48 64 4C 6F 47 20  A7
6449 : D6 09 C4 45 52 20 C7 45  4D
6451 : 47 45 4E 53 54 41 4E 44  4A
6459 : 20 49 53 54 20 4E 49 43  9D
6461 : 48 54 20 49 4E 20 44 45  86
6469 : 52 20 CB 49 53 54 45 2E  31
6471 : 20 20 20 0D 00 4C 13 35  64
6479 : AD 1E 24 C9 01 F0 2D 20  04
6481 : D6 09 D3 49 45 20 53 45  27
6489 : 48 45 4E 20 45 49 4E 45  6E
6491 : 20 47 45 53 43 48 4C 4F  57
6499 : 53 53 45 4E 45 20 CB 49  CB
64A1 : 53 54 45 2E 20 20 0D 0D  87
64A9 : 00 4C 13 35 20 D6 09 D3  BF
64B1 : 49 45 20 53 45 48 45 4E  58
64B9 : 20 45 49 4E 45 20 4F 46  B7
64C1 : 46 45 4E 45 20 CB 49 53  12
64C9 : 54 45 2E 20 20 20 20 20  13
64D1 : 20 20 20 20 20 20 20 C4  1B
64D9 : 41 52 49 4E 20 53 45 48  A2
64E1 : 45 4E 20 53 49 45 3A 20  A8
64E9 : 20 00 A9 00 8D 09 0B A9  14
64F1 : 01 8D 3B 32 8D 3C 32 20  92
64F9 : 3E 32 4C 13 35 20 D6 09  88
6501 : C4 49 45 20 CB 49 53 54  BC
6509 : 45 20 49 53 54 20 47 45  09
6511 : 53 43 48 4C 4F 53 53 45  09
6519 : 4E 2E 20 20 20 20 20 0D  28
6521 : 0D 00 4C 13 35 20 19 35  C7
6529 : 0D FD 09 0D FF 09 F0 01  27
6531 : 60 AD FE 09 C9 01 D0 F8  22
6539 : AE FC 09 F0 F3 BD D5 23  90
6541 : C9 01 D0 47 BD EB 23 C9  03
6549 : 03 D0 40 AD 1E 24 C9 01  A6
6551 : F0 03 4C 8F 65 20 D6 09  8D
6559 : CF 4B 2E 20 D3 49 45 20  3A
6561 : 4C 45 47 45 4E 20 45 54  6E
6569 : 57 41 53 20 49 4E 20 44  4A
6571 : 49 45 20 CB 49 53 54 45  E9
6579 : 2E 20 20 20 20 0D 0D 00  62
6581 : A9 01 9D D5 23 9D EB 23  E2
6589 : 4C 13 35 4C 46 4D 20 D6  33
6591 : 09 C4 49 45 20 CB 49 53  24
6599 : 54 45 20 49 53 54 20 47  A8
65A1 : 45 53 43 48 4C 4F 53 53  9D
65A9 : 45 4E 21 20 20 20 20 20  26
65B1 : 0D 0D 00 4C 13 35 20 19  5C
65B9 : 35 0D FD 09 0D FF 09 F0  EC
65C1 : 01 60 AD FE 09 C9 01 D0  C2
65C9 : F8 AE FC 09 F0 F3 BD D5  CA
65D1 : 23 C9 01 D0 46 BD EB 23  7B
65D9 : C9 01 D0 3F AD 1E 24 C9  2F
65E1 : 01 F0 03 4C 1F 66 20 D6  F8
65E9 : 09 D3 49 45 20 4E 45 48  F1
65F1 : 4D 45 4E 20 45 54 57 41  4F
65F9 : 53 20 41 55 53 20 [illegible] 45  29
6601 : 52 20 CB 49 53 54 45 2E  C9
6609 : 20 20 20 0D 0D 00 A9 01  5D
6611 : 9D D5 23 A9 03 9D EB 23  AA
6619 : 4C 13 35 4C D1 4D 20 D6  7B
6621 : 09 C4 49 45 20 CB 49 53  B4
6629 : 54 45 20 49 53 54 20 47  38
6631 : 45 53 43 48 4C 4F 53 53  2D
6639 : 45 4E 2E 20 20 20 20 20  F9
6641 : 0D 0D 00 4C 13 35 20 19  EC
6649 : 35 0D FC 09 0D FD 09 0D  64
6651 : FF 09 F0 01 60 AD FE 09  B3
6659 : C9 01 D0 F8 EA EA EA AD  03
6661 : DD 23 C9 01 D0 07 AD F3  46
6669 : 23 C9 03 F0 40 20 D6 09  C2
6671 : D3 49 45 20 48 41 42 45  60
6679 : 4E 20 4E 49 43 48 54 20  9C
6681 : 44 45 4E 20 50 41 53 53  02
6689 : 45 4E 44 45 4E 20 20 20  56
6691 : 20 20 20 20 20 20 20 20  91
6699 : D3 43 48 4C 55 45 53 53  1D
66A1 : 45 4C 21 20 20 20 20 0D  F6
66A9 : 0D 00 4C 13 35 AD 1E 24  AD
66B1 : D0 25 20 D6 09 C4 49 45  5D
66B9 : 20 CB 49 53 54 45 20 57  1A
66C1 : 41 52 20 42 45 52 45 49  0A
66C9 : 54 53 20 5A 55 2E 20 20  A2
66D1 : 20 0D 0D 00 4C 13 35 A9  41
66D9 : 00 8D 1E 24 20 D6 09 D3  30
66E1 : 49 45 20 53 43 48 4C 49  7A
66E9 : 45 53 53 45 4E 20 44 49  DF
66F1 : 45 20 CB 49 53 54 45 2E  AC
66F9 : 20 A0 A0 0D 0D 00 4C 13  5B
6701 : 35 20 19 35 0D FC 09 0D  2A
6709 : FD 09 0D FF 09 F0 01 60  AB
6711 : AD FE 09 C9 01 D0 F8 EA  09
6719 : EA EA AD DD 23 C9 01 D0  C5
6721 : 07 AD F3 23 C9 03 F0 03  DF
6729 : 4C 6E 66 AD 1E 24 C9 01  28
6731 : D0 29 20 D6 09 C4 49 45  DF
6739 : 20 CB 49 53 54 45 20 49  7E
6741 : 53 54 20 42 45 52 45 49  9D
6749 : 54 53 20 4F 46 46 45 4E  81
6751 : 2E 20 20 20 20 0D 0D 00  3A
6759 : 4C 13 35 A9 01 8D 1E 24  EF
6761 : 20 D6 09 D3 49 45 20 4F  87
6769 : 45 46 46 4E 45 4E 20 44  FD
6771 : 49 45 20 CB 49 53 54 45  E9
6779 : 2E 20 A0 20 20 20 20 0D  81
6781 : 0D 00 4C 13 35 C9 43 D0  54
6789 : 03 4C 37 50 C9 44 F0 F9  01
6791 : C9 45 F0 03 4C BF 68 AD  59
6799 : FB 09 C9 0C D0 03 4C 1E  9F
67A1 : 41 C9 15 D0 03 4C DE 67  03
67A9 : C9 16 D0 22 20 AF 45 20  CB
67B1 : D6 09 D7 41 53 53 45 52  B3
67B9 : 20 50 4C 41 45 54 53 43  07
67C1 : 48 45 52 54 21 20 20 20  9F
67C9 : 0D 0D 00 4C 13 35 C9 19  1B
67D1 : D0 03 4C 2B 68 C9 1D D0  86
67D9 : 03 4C 8A 68 60 20 19 35  88
67E1 : 0D FC 09 0D FD 09 0D FF  AD
67E9 : 09 F0 01 60 AD FE 09 C9  41
67F1 : 05 F0 03 4C D8 42 20 D6  86
67F9 : 09 D3 49 45 20 53 45 48  29
6801 : 45 4E 20 54 49 45 46 20  18
6809 : 55 4E 54 45 4E 20 D7 41  0B
6811 : 53 53 45 52 20 49 4D 20  6B
6819 : C2 52 55 4E 4E 45 4E 2E  C8
6821 : 20 20 20 20 20 0D 00 4C  61
6829 : 13 35 20 19 35 0D FD 09  C8
6831 : 0D FF 09 F0 01 60 AD FE  66
6839 : 09 C9 05 D0 F8 AE FC 09  8D
6841 : F0 F3 BD D5 23 C9 01 D0  7B
6849 : 3D BD EB 23 C9 03 D0 36  28
6851 : 20 D6 09 D3 49 45 20 57  87
6859 : 45 52 46 45 4E 20 45 54  A5
6861 : 57 41 53 20 49 4E 20 44  42
6869 : 45 4E 20 C2 52 55 4E 4E  DB
6871 : 45 4E 2E 20 20 20 0D 0D  BE
6879 : 00 A9 05 9D D5 23 A9 01  62
6881 : 9D EB 23 4C 13 35 4C 46  FF
6889 : 4D 20 19 35 0D FD 09 0D  D2
6891 : FF 09 F0 01 60 AD FE 09  F3
6899 : C9 05 D0 F8 20 D6 09 C4  9E
68A1 : 41 53 20 57 41 45 52 45  91
68A9 : 20 D3 45 4C 42 53 54 4D  38
68B1 : 4F 52 44 21 20 20 20 20  22
68B9 : 0D 0D 00 4C 13 35 C9 49  6B
68C1 : B0 03 4C 37 50 C9 49 F0  47
68C9 : 03 4C 57 69 AD FB 09 C9  68
68D1 : 16 D0 28 20 AF 45 20 D6  B1
68D9 : 09 D3 49 45 20 48 4F 45  D3
68E1 : 52 45 4E 20 56 49 45 4C  CB
68E9 : 45 20 D3 54 49 4D 4D 45  7C
68F1 : 4E 21 20 20 20 0D 0D 00  7B
68F9 : 4C 13 35 C9 0C D0 03 4C  41
6901 : 1E 41 C9 15 D0 03 4C D8  DD
6909 : 42 C9 0B 90 01 60 C9 03  45
6911 : F0 FB 20 AF 45 20 D6 09  C0
6919 : D3 49 45 20 57 55 52 44  D8
6921 : 45 4E 20 53 4F 45 42 45  B3
6929 : 4E 20 41 55 46 20 44 45  B3
6931 : 52 20 C6 4C 55 43 48 54  08
6939 : 20 20 20 20 20 20 20 20  39
6941 : 45 52 53 43 48 4F 53 53  E0
6949 : 45 4E 2E 2E 2E 20 0D 0D  39
6951 : 00 68 68 4C E3 3E C9 4A  15
6959 : D0 03 4C 37 50 C9 56 90  73
6961 : F9 C9 56 D0 03 4C 74 69  26
6969 : C9 57 F0 F9 C9 59 F0 F5  70
6971 : 4C B1 69 AD FB 09 C9 0C  ED
6979 : D0 03 4C 1E 41 C9 15 D0  FA
6981 : 03 4C D8 42 C9 16 D0 27  08
6989 : 20 AF 45 20 D6 09 C5 53  4A
6991 : 20 49 53 54 20 41 42 53  71
6999 : 4F 4C 55 54 20 53 54 49  6F
69A1 : 4C 4C 21 21 21 20 20 20  54
69A9 : 20 0D 0D 00 4C 13 35 60  86
69B1 : C9 58 D0 03 4C 37 50 C9  8E
69B9 : 5A F0 F9 C9 5B F0 03 4C  25
69C1 : 24 6A AD FB 09 C9 0C D0  B6
69C9 : 2B 20 AF 45 20 D6 09 C2  FB
69D1 : 45 53 4F 52 47 45 4E 20  F6
69D9 : 53 49 45 20 53 49 43 48  43
69E1 : 20 45 49 4E 20 D4 49 43  14
69E9 : 4B 45 54 21 20 20 0D 0D  61
69F1 : 00 4C 13 35 C9 15 D0 03  11
69F9 : 4C D8 42 C9 16 F0 01 60  29
6A01 : 20 AF 45 20 D6 09 D3 49  E6
6A09 : 45 20 48 4F 45 52 45 4E  F3
6A11 : 20 C6 4C 55 47 5A 45 55  59
6A19 : 47 45 2E 20 20 0D 0D 00  31
6A21 : 4C 13 35 C9 5C F0 9B C9  CD
6A29 : 5D F0 97 C9 5E F0 03 4C  2F
6A31 : 69 6A AD FB 09 C9 0C D0  6B
6A39 : 03 4C 1E 41 C9 15 D0 03  A1
6A41 : 4C D8 42 C9 16 F0 01 60  71
6A49 : 20 AF 45 20 D6 09 D3 49  2E
6A51 : 45 20 48 4F 45 52 45 4E  3B
6A59 : 20 CD 55 53 49 4B 2E 20  08
6A61 : 20 20 0D 0D 00 4C 13 35  8F
6A69 : C9 60 90 01 60 4C 33 6A  B1
6A71 : EA EA EA EA EA EA EA EA  70
6A79 : EA EA EA EA EA EA 60 EA  4E
6A81 : EA EA 20 3F 6C 20 49 6C  96
6A89 : 20 AC 6C 20 E1 6C 20 20  61
6A91 : 6D 20 5A 6D 20 8F 6D 20  C7
6A99 : E5 6E 20 D5 6E 20 A7 6F  DD
6AA1 : 20 B3 70 20 19 71 20 46  E5
```

## Listing »Freiheit« (Fortsetzung)

```
6AA9 : 71 20 EC 71 20 30 72 20  21
6AB1 : D2 72 20 0D 73 20 7D 75  7F
6AB9 : 20 AB 74 20 0C 75 20 10  DD
6AC1 : 75 20 95 73 20 57 76 20  F1
6AC9 : 91 76 20 E8 76 20 D5 6A  4F
6AD1 : EA EA EA 60 4C 54 6B C9  A0
6AD9 : 01 D0 22 20 D6 09 D3 49  67
6AE1 : 45 20 53 49 4E 44 20 56  69
6AE9 : 45 52 48 55 4E 47 45 52  ED
6AF1 : 54 2E 2E 2E 0D 00 68 68  F1
6AF9 : 68 68 4C E3 3E C9 0B B0  E5
6B01 : 2D 20 D6 09 D3 49 45 20  F2
6B09 : 53 49 4E 44 20 4B 55 52  73
6B11 : 5A 20 56 4F 52 20 44 45  BD
6B19 : 4D 20 D6 45 52 48 55 4E  2E
6B21 : 47 45 52 4E 21 20 20 20  3D
6B29 : 0D 0D 00 4C 4E 6B 20 D6  B5
6B31 : 09 D3 49 45 20 53 49 4E  7D
6B39 : 44 20 53 45 48 52 20 48  33
6B41 : 55 4E 47 52 49 47 21 20  6D
6B49 : 20 20 0D 0D 00 CE 77 24  FB
6B51 : 4C 5C 6B AD 77 24 F0 F8  AA
6B59 : 4C D8 6A EA EA EA AD D3  6E
6B61 : 23 C9 5E F0 01 60 AD D4  92
6B69 : 23 C9 00 D0 F8 AD FB 09  8A
6B71 : C9 12 D0 F1 68 68 68 68  F2
6B79 : 4C 81 3E A9 0D 20 D2 FF  68
6B81 : 4C 80 6A 20 BC 3D 4C AF  F2
6B89 : 29 A2 02 A9 FF 9D 76 24  C8
6B91 : 20 D6 09 C5 44 45 20 53  AD
6B99 : 41 47 54 2C 20 45 52 20  CE
6BA1 : 57 45 52 44 45 20 42 41  99
6BA9 : 4C 44 20 45 4E 54 4C 41  03
6BB1 : 53 53 45 4E 2E 0D D7 45  FE
6BB9 : 4E 4E 20 53 49 45 20 46  6D
6BC1 : 4C 49 45 48 45 4E 20 57  02
6BC9 : 4F 4C 4C 54 45 4E 2C 20  94
6BD1 : 53 4F 4C 4C 54 45 4E 20  51
6BD9 : 53 49 45 0D 56 45 52 53  43
6BE1 : 55 43 48 45 4E 2C 20 41  DC
6BE9 : 55 46 20 44 49 45 20 CE  CF
6BF1 : 41 43 48 42 41 52 49 4E  97
6BF9 : 53 45 4C 20 5A 55 0D 47  19
6C01 : 45 4C 41 4E 47 45 4E 20  9F
6C09 : 55 4E 44 20 44 4F 52 54  4B
6C11 : 20 45 49 4E 20 C6 4C 55  04
6C19 : 47 5A 45 55 47 20 4E 45  C3
6C21 : 48 4D 45 4E 2E 0D 0D C5  36
6C29 : 44 45 20 53 43 48 4C 55  D5
6C31 : 4D 4D 45 52 54 20 45 49  AE
6C39 : 4E 2E 0D 0D 00 60 A2 02  15
6C41 : BD 76 24 D0 F8 4C 8A 6B  4F
6C49 : A2 03 BD 5D 24 A8 C8 B9  A6
6C51 : 9A 6C D0 05 A0 00 4C 50  D2
6C59 : 6C 9D 2B 24 98 9D 5D 24  17
6C61 : AD D4 23 F0 01 60 AD D3  D1
6C69 : 23 DD 2B 24 D0 F7 20 D6  C5
6C71 : 09 C4 45 52 20 D7 41 45  C8
6C79 : 52 54 45 52 20 48 41 54  83
6C81 : 20 53 49 45 20 45 52 57  6A
6C89 : 49 53 43 48 54 2E 2E 2E  22
6C91 : 0D 00 68 68 68 68 4C E3  88
6C99 : 3E 05 06 04 06 09 06 05  27
6CA1 : 05 05 05 05 05 05 00 00  83
6CA9 : 00 00 00 A2 04 BD 5D 24  EA
6CB1 : A8 C8 B9 D4 6C D0 04 A0  65
6CB9 : 00 F0 F7 9D 2B 24 98 9D  54
6CC1 : 5D 24 AD D4 23 F0 01 60  B5
6CC9 : AD D3 23 DD 2B 24 D0 F7  EB
6CD1 : 4C 6F 6C 03 03 07 05 05  D7
6CD9 : 05 05 07 03 00 00 00 00  83
6CE1 : A2 05 BD 5D 24 A8 C8 B9  3F
6CE9 : 09 6D D0 04 A0 00 F0 F7  1B
6CF1 : 9D 2B 24 98 9D 5D 24 AD  F1
6CF9 : D4 23 F0 01 60 AD D3 23  C4
6D01 : DD 2B 24 D0 F7 4C 6F 6C  0F
6D09 : 0A 0A 0A 0A 0A 0A 0A 0A  09
6D11 : 08 0C 0C 0C 0C 0C 0C 0C  0D
6D19 : 0C 08 00 00 00 00 00 A2  6F
6D21 : 06 BD 5D 24 A8 C8 B9 48  2A
6D29 : 6D D0 04 A0 00 F0 F7 9D  B6
6D31 : 2B 24 98 9D 5D 24 AD D4  9F
6D39 : 23 F0 01 60 AD D3 23 DD  E2
6D41 : 2B 24 D0 F7 4C 6F 6C 0C  BB
6D49 : 0C 0C 0C 18 17 1B 1E 20  65
6D51 : 1E 1B 17 18 00 00 00 00  C6
6D59 : 00 A2 07 BD 5D 24 A8 C8  4F
6D61 : B9 82 6D D0 04 A0 00 F0  F8
6D69 : F7 9D 2B 24 98 9D 5D 24  B2
6D71 : AD D4 23 F0 01 60 AD D3  E1
6D79 : 23 DD 2B 24 D0 F7 4C 6F  B7
6D81 : 6C 0F 10 11 13 14 13 11  DB
6D89 : 10 00 00 00 00 00 A2 08  34
6D91 : BD 5D 24 A8 C8 B9 B7 6D  2F
6D99 : D0 04 A0 00 F0 F7 9D 2B  2F
6DA1 : 24 98 9D 5D 24 AD D4 23  6E
6DA9 : F0 01 60 AD D3 23 DD 2B  0C
6DB1 : 24 D0 F7 4C 6F 6C 22 23  EE
6DB9 : 00 00 00 00 00 00 00 A2  FF
6DC1 : 09 BD 76 24 C9 00 D0 43  31
6DC9 : 20 D6 09 C4 45 52 20 CB  2E
6DD1 : 45 4C 4C 4E 45 52 20 53  27
6DD9 : 41 47 54 3A 20 27 C5 49  FF
6DE1 : 4E 20 53 43 48 4F 45 4E  2D
6DE9 : 45 52 20 D4 41 47 20 20  09
6DF1 : 20 20 20 48 45 55 54 45  0D
6DF9 : 2C 20 4E 49 43 48 54 20  FA
6E01 : 57 41 48 52 3F 27 20 20  43
6E09 : 0D 0D 00 A2 09 BD 76 24  92
6E11 : C9 01 D0 2E 20 D6 09 C4  BB
6E19 : 45 52 20 CB 45 4C 4C 4E  8D
6E21 : 45 52 20 53 41 47 54 3A  16
6E29 : 20 27 D3 49 45 20 57 55  58
6E31 : 45 4E 53 43 48 45 4E 3F  41
6E39 : 27 20 20 20 20 20 20 20  40
6E41 : 0D 00 A2 09 BD 76 24 18  68
6E49 : 69 01 C9 04 90 02 A9 00  E5
6E51 : 9D 76 24 EA AD DE 23 C9  82
6E59 : 01 D0 35 AD F4 23 C9 03  5B
6E61 : D0 2E 20 D6 09 C4 45 52  9C
6E69 : 20 CB 45 4C 4C 4E 45 52  3B
6E71 : 20 53 41 47 54 3A 20 27  5A
6E79 : C7 55 54 45 4E 20 C1 50  36
6E81 : 50 45 54 49 54 21 27 20  DD
6E89 : 20 20 20 20 20 20 0D 00  FD
6E91 : A2 15 BD D5 23 C9 09 D0  2E
6E99 : 3A BD EB 23 C9 03 D0 33  6F
6EA1 : 20 35 77 EA EA 20 D6 09  B4
6EA9 : C4 45 52 20 CB 45 4C 4C  59
6EB1 : 4E 45 52 20 47 49 42 54  AB
6EB9 : 20 49 48 4E 45 4E 20 45  2C
6EC1 : 54 57 41 53 20 5A 55 20  E6
6EC9 : 45 53 53 45 4E 2E 20 20  4C
6ED1 : 20 0D 00 60 4C F6 6E A2  FF
6ED9 : 09 A9 01 9D D5 23 A9 03  CE
6EE1 : 9D EB 23 60 AD D3 23 C9  E2
6EE9 : 26 D0 F8 AD D4 23 C9 00  F9
6EF1 : D0 F1 4C C0 6D AD D3 23  BF
6EF9 : C9 2F D0 E7 AD D4 23 C9  2C
6F01 : 00 D0 E0 20 D6 09 C4 45  F9
6F09 : 52 20 CB 4C 45 49 44 45  22
6F11 : 52 56 45 52 4B 41 45 55  A8
6F19 : 46 45 52 20 53 41 47 54  9F
6F21 : 3A 20 27 C9 43 48 20 48  F6
6F29 : 41 42 45 20 20 20 47 45  8B
6F31 : 4E 41 55 20 44 41 53 20  55
6F39 : D2 49 43 48 54 49 47 45  C1
6F41 : 20 46 55 45 52 20 53 49  88
6F49 : 45 21 20 20 20 20 20 0D  C9
6F51 : 0D 00 A2 07 BD D5 23 C9  92
6F59 : 0A F0 01 60 BD EB 23 C9  83
6F61 : 03 D0 F8 20 D6 09 C4 45  62
6F69 : 52 20 CB 4C 45 49 44 45  82
6F71 : 52 56 45 52 4B 41 45 55  08
6F79 : 46 45 52 20 47 49 42 54  6B
6F81 : 20 49 48 4E 45 4E 20 45  F4
6F89 : 49 4E 45 4E 20 20 C1 4E  BB
6F91 : 5A 55 47 2E 20 20 0D 0D  7F
6F99 : 00 A2 06 A9 01 9D D5 23  3C
6FA1 : A9 03 4C 5D 77 EA A2 0B  FA
6FA9 : BD 5D 24 A8 C8 B9 BA 6F  57
6FB1 : D0 04 A0 00 F0 F7 4C C2  31
6FB9 : 6F 28 28 28 28 00 00 00  CE
6FC1 : 00 9D 2B 24 98 9D 5D 24  13
6FC9 : EA EA EA AD D3 23 DD 2B  BD
6FD1 : 24 F0 01 60 AD D4 23 DD  83
6FD9 : 44 24 D0 F7 A2 02 BD D5  3F
6FE1 : 23 C9 0B D0 44 BD EB 23  EE
6FE9 : C9 03 D0 3D 20 54 77 20  D2
6FF1 : D6 09 C4 45 52 20 C6 41  E9
6FF9 : 45 4C 53 43 48 45 52 20  DA
7001 : 47 49 42 54 20 49 48 4E  12
7009 : 45 4E 20 45 49 4E 45 4E  DF
7011 : 20 D0 41 53 53 2E 20 20  BC
7019 : 20 20 20 20 0D 00 A9 01  CF
7021 : 8D D6 23 A9 03 8D EC 23  AE
7029 : EA 4C 46 77 C9 1B F0 01  F5
7031 : 60 AD 01 0A C9 01 D0 F8  C3
7039 : AD 02 0A C9 00 D0 F1 AD  4D
7041 : 03 0A C9 00 D0 EA AD 04  DF
7049 : 0A C9 01 D0 E3 EA A2 02  B6
7051 : BD D5 23 C9 0B D0 08 BD  CE
7059 : EB 23 C9 03 D0 01 60 20  7F
7061 : D6 09 C4 45 52 20 C6 41  59
7069 : 45 4C 53 43 48 45 52 20  4A
7071 : 53 41 47 54 2C 20 53 49  65
7079 : 45 20 53 4F 4C 4C 45 4E  66
7081 : 20 49 48 4D 20 45 52 53  1E
7089 : 54 20 53 45 49 4E 45 20  C7
7091 : 56 45 52 4C 4F 52 45 4E  E1
7099 : 45 20 D5 48 52 20 46 49  3F
70A1 : 4E 44 45 4E 2E 20 20 20  D1
70A9 : 20 20 20 0D 0D 00 EA EA  D5
70B1 : EA 60 A2 0C EA EA EA A9  FA
70B9 : 3C EA EA 9D 2B 24 CD D3  8B
70C1 : 23 F0 01 60 BD 44 24 CD  D3
70C9 : D4 23 D0 F7 EA EA EA 20  54
70D1 : D6 09 C4 45 52 20 C2 41  B9
70D9 : 55 45 52 20 48 55 53 54  8F
70E1 : 45 54 20 46 55 45 52 43  71
70E9 : 48 54 45 52 4C 49 43 48  A4
70F1 : 2E 20 20 20 0D 0D 00 A2  BA
70F9 : 05 BD D5 23 C9 0C F0 01  79
7101 : 60 BD EB 23 C9 03 D0 F8  89
7109 : A9 01 A2 04 9D D5 23 A9  C4
7111 : 03 9D EB 23 EA 4C 6B 77  F0
7119 : A2 0D 20 41 71 C8 B9 2B  0D
7121 : 71 D0 04 A0 00 D0 F7 4C  0E
7129 : 39 71 42 25 27 2B 30 32  41
7131 : 30 2B 27 25 00 00 00 00  65
7139 : 9D 2B 24 98 9D 5D 24 60  9E
7141 : BD 5D 24 A8 60 A2 0E BD  9A
7149 : 76 24 C9 37 F0 F6 AD D3  50
7151 : 23 C9 4B D0 EF AD D4 23  4C
7159 : C9 00 D0 E8 A9 49 8D D3  36
7161 : 23 EA EA EA 20 D6 09 C5  7A
7169 : 49 4E 20 D0 4F 4C 49 5A  2D
7171 : 49 53 54 20 48 41 54 20  9D
7179 : 53 49 45 20 4E 41 43 48  53
7181 : 20 D3 55 45 44 45 4E 20  71
7189 : 5A 55 20 0D 45 49 4E 45  9A
7191 : 52 20 D0 4F 4C 49 5A 45  14
7199 : 49 4B 4F 4E 54 52 4F 4C  D3
71A1 : 4C 45 20 4D 49 54 47 45  20
71A9 : 4E 4F 4D 4D 45 4E 2C 0D  2D
71B1 : 45 53 20 53 49 4E 44 20  6B
71B9 : 41 42 45 52 20 4E 4F 43  EF
71C1 : 48 20 D0 45 52 53 4F 4E  90
71C9 : 45 4E 20 56 4F 52 20 49  A3
71D1 : 48 4E 45 4E 0D 41 4E 20  B0
71D9 : 44 45 52 20 D2 45 49 48  65
71E1 : 45 2E 20 20 20 0D 0D 00  E8
71E9 : 4C C8 72 AD D3 23 C9 49  FC
71F1 : D0 3C AD D4 23 C9 00 D0  07
71F9 : 35 20 D6 09 C5 49 4E 20  35
7201 : D0 4F 4C 49 5A 49 53 54  9B
7209 : 20 52 45 44 45 54 20 4D  3E
7211 : 49 54 20 45 49 4E 45 4D  EC
7219 : 20 D0 41 53 53 41 4E 54  7D
7221 : 45 4E 2E 20 20 20 20 20  E1
7229 : 20 20 20 0D 0D 00 60 A2  9B
7231 : 10 BD 5D 24 A8 C8 B9 43  3A
7239 : 72 D0 04 A0 00 F0 F7 4C  28
7241 : 55 72 29 3A 3A 3A 29 25  C5
7249 : 29 29 29 29 29 29 29 00  F7
7251 : 00 00 00 00 9D 2B 24 98  46
7259 : 9D 5D 24 EA AD D3 23 DD  CD
7261 : 2B 24 F0 01 60 AD D4 23  08
7269 : DD 44 24 D0 F7 A2 06 BD  B4
7271 : D5 23 C9 01 D0 07 BD EB  7E
7279 : 23 C9 03 F0 E7 20 D6 09  4D
7281 : D3 49 45 20 57 45 52 44  C0
7289 : 45 4E 20 56 4F 4E 20 45  3B
7291 : 49 4E 45 52 20 D3 54 52  34
7299 : 45 49 46 45 20 41 55 46  AB
72A1 : 20 44 45 52 20 20 20 20  43
72A9 : C6 4C 55 43 48 54 20 45  85
72B1 : 52 53 43 48 4F 53 53 45  EE
72B9 : 4E 2E 20 20 20 20 0D 00  62
72C1 : 68 68 68 68 4C E3 3E A9  B5
72C9 : 37 A2 0E 9D 76 24 4C B7  B2
72D1 : 3D A2 11 BD 5D 24 A8 C8  87
72D9 : B9 E5 72 D0 04 A0 00 F0  63
72E1 : F7 4C 00 73 44 43 3E 3D  3F
72E9 : 3A 29 29 29 29 29 29 29  FA
72F1 : 29 3A 3D 3E 43 00 00 00  83
72F9 : 00 00 00 00 00 00 00 9D  35
7301 : 2B 24 98 9D 5D 24 EA EA  91
7309 : EA 4C 5D 72 A2 12 BD 5D  2B
7311 : 24 A8 C8 B9 20 73 D0 04  DC
7319 : A0 00 F0 F7 4C 2B 73 46  6D
7321 : 47 48 43 48 47 00 00 00  DB
7329 : 00 00 9D 2B 24 98 9D 5D  2E
7331 : 24 EA EA EA BD 2B 24 CD  44
7339 : D3 23 F0 01 60 BD 44 24  47
7341 : CD D4 23 D0 F7 20 D6 09  49
7349 : C4 45 52 20 C6 4F 45 52  E9
7351 : 53 54 45 52 20 53 41 47  9A
7359 : 54 2C 20 45 53 20 57 49  9A
7361 : 4D 4D 4C 45 20 49 4E 20  D6
7369 : 44 45 52 20 20 20 20 20  AC
7371 : C7 45 47 45 4E 44 20 4E  79
7379 : 55 52 20 53 4F 20 56 4F  58
7381 : 4E 20 D0 4F 4C 49 5A 49  08
7389 : 53 54 45 4E 2E 20 20 20  C6
7391 : 0D 0D 00 60 AD D3 23 C9  CB
7399 : 5E D0 F8 AD D4 23 C9 00  E1
73A1 : D0 F1 A2 17 BD 76 24 C9  A9
73A9 : 47 D0 E8 A2 13 BD 76 24  28
73B1 : D0 47 EA EA EA 20 D6 09  5A
73B9 : C4 49 45 20 D3 54 45 57  1B
73C1 : 41 52 44 45 53 53 20 53  DC
73C9 : 41 47 54 2C 20 53 49 45  95
73D1 : 20 53 4F 4C 4C 45 4E 20  61
73D9 : 49 48 52 20 44 41 53 20  BB
```

```
73E1 : D4 49 43 4B 45 54 20 47  9A
73E9 : 45 42 45 4E 2E 20 20 20  0F
73F1 : 0D 0D 00 A9 01 9D 76 24  D9
73F9 : 60 C9 01 D0 FB A2 03 BD  F5
7401 : D5 23 C9 13 D0 73 BD EB  B4
7409 : 23 C9 03 D0 6C A9 01 9D  3F
7411 : D5 23 20 D6 09 D3 49 45  3A
7419 : 20 42 45 4B 4F 4D 4D 45  34
7421 : 4E 20 49 48 52 20 D4 49  E7
7429 : 43 4B 45 54 20 5A 55 52  BD
7431 : 55 45 43 4B 2E 0D C4 49  54
7439 : 45 20 D3 54 45 57 41 52  C6
7441 : 44 45 53 53 20 57 55 45  04
7449 : 4E 53 43 48 54 20 49 48  17
7451 : 4E 45 4E 20 45 49 4E 45  3C
7459 : 4E 20 20 20 20 20 47 55  8E
7461 : 54 45 4E 20 C6 4C 55 47  A2
7469 : 2E 20 20 20 20 20 0D 0D  05
7471 : 00 A9 FF A2 13 9D 76 24  DA
7479 : 60 20 D6 09 CD 41 4E 20  20
7481 : 4C 49 45 46 45 52 54 20  04
7489 : 53 49 45 20 41 4E 20 44  66
7491 : 49 45 20 D0 4F 4C 49 5A  D0
7499 : 45 49 20 41 55 53 2E 2E  B8
74A1 : 2E 0D 00 68 68 68 68 4C  67
74A9 : E3 3E EA EA EA A2 04 BD  12
74B1 : D5 23 C9 14 F0 01 60 BD  21
74B9 : EB 23 C9 03 D0 F8 A2 03  6E
74C1 : A9 01 9D D5 23 A9 03 9D  D4
74C9 : EB 23 20 76 77 20 D6 09  03
74D1 : D3 49 45 20 45 52 48 41  29
74D9 : 4C 54 45 4E 20 49 48 52  7D
74E1 : 20 D4 49 43 4B 45 54 20  97
74E9 : 46 55 45 52 20 44 45 4E  4B
74F1 : 20 C6 4C 55 47 20 20 20  68
74F9 : 4E 41 43 48 20 C8 41 55  BA
7501 : 53 45 21 20 20 20 20 0D  E1
7509 : 0D 00 60 A2 15 D0 02 A2  A8
7511 : 16 BD 2B 24 CD D3 23 F0  3F
7519 : 01 60 BD 44 24 CD D4 23  8C
7521 : D0 F7 A2 06 BD D5 23 C9  01
7529 : 01 D0 08 BD EB 23 C9 03  51
7531 : D0 01 60 20 D6 09 C5 49  FD
7539 : 4E 20 C6 4C 55 47 48 41  06
7541 : 46 45 4E 50 4F 4C 49 5A  F9
```

```
7549 : 49 53 54 20 48 41 54 20  75
7551 : 53 49 45 20 41 4C 53 20  A2
7559 : 20 20 20 20 20 20 D3 54  90
7561 : 52 41 45 46 4C 49 4E 47  45
7569 : 20 45 52 4B 41 4E 4E 54  92
7571 : 2E 2E 2E 0D 00 68 68 68  99
7579 : 68 4C E3 3E AD D3 23 C9  62
7581 : 5E F0 01 60 AD D4 23 C9  45
7589 : 00 D0 F8 A2 17 BD 76 24  05
7591 : C9 47 D0 01 60 C9 00 D0  48
7599 : 39 20 D6 09 C4 45 52 20  B9
75A1 : DA 4F 45 4C 4C 4E 45 52  EF
75A9 : 20 4D 4F 45 43 48 54 45  3F
75B1 : 20 49 48 52 45 4E 20 D0  BB
75B9 : 41 53 53 20 48 41 42 45  9F
75C1 : 4E 2E 20 20 20 20 20 0D  D0
75C9 : 00 A9 01 9D 76 24 EA EA  9C
75D1 : EA 60 A2 01 BD D5 23 C9  5F
75D9 : 17 D0 41 BD EB 23 C9 03  65
75E1 : D0 3A A9 01 9D D5 23 20  AE
75E9 : D6 09 D3 49 45 20 42 45  4B
75F1 : 4B 4F 4D 4D 45 4E 20 49  BB
75F9 : 48 52 45 4E 20 D0 41 53  BA
7601 : 53 20 5A 55 52 55 45 43  11
7609 : 4B 2E 20 20 20 20 0D 0D  C9
7611 : 00 A2 17 A9 47 9D 76 24  E1
7619 : EA EA EA 60 20 D6 09 C4  A5
7621 : 45 52 20 DA 4F 45 4C 4C  DC
7629 : 4E 45 52 20 4C 49 45 46  63
7631 : 45 52 54 20 53 49 45 20  8D
7639 : 44 45 52 20 D0 4F 4C 49  04
7641 : 5A 45 49 20 20 20 20 41  9A
7649 : 55 53 2E 2E 2E 0D 00 68  B5
7651 : 68 68 68 4C E3 3E A2 18  7C
7659 : BD 5D 24 A8 C8 C0 03 D0  23
7661 : 02 A0 00 98 9D 5D 24 AD  77
7669 : D4 23 F0 01 60 AD D3 23  34
7671 : C9 24 90 F8 C9 35 B0 F4  82
7679 : C0 00 F0 01 60 EA EA EA  74
7681 : AD D3 23 9D 2B 24 AD D4  C9
7689 : 23 9D 44 24 EA EA EA 60  83
7691 : A2 0D BD D5 23 C9 01 F0  4A
7699 : 01 60 BD EB 23 C9 03 D0  E5
76A1 : F8 EA EA EA 20 D6 09 C4  8D
76A9 : 49 45 20 C1 4C 41 52 4D  88
```

```
76B1 : 41 4E 4C 41 47 45 20 48  04
76B9 : 41 54 20 44 49 45 20 D0  96
76C1 : 4F 4C 49 5A 45 49 20 20  33
76C9 : 20 20 20 20 20 20 20 41  0C
76D1 : 4C 41 52 4D 49 45 52 54  AD
76D9 : 2E 2E 2E 20 20 20 0D 00  E5
76E1 : 68 68 68 68 4C E3 3E AD  DD
76E9 : D3 23 C9 2E F0 01 60 AD  7A
76F1 : D4 23 C9 00 D0 FB A9 2C  9D
76F9 : 8D D3 23 EA EA EA 20 D6  CA
7701 : 09 D3 49 45 20 53 49 4E  4D
7709 : 44 20 57 49 45 44 45 52  8D
7711 : 20 41 55 46 20 44 45 52  CE
7719 : 20 D3 54 52 41 53 53 45  09
7721 : 2E 20 20 20 20 0D 0D 00  0A
7729 : EA EA EA EA 4C B7 3D 68  E8
7731 : 68 4C AF 29 A9 00 9D D5  8D
7739 : 23 9D EB 23 4C D8 6E 20  10
7741 : D5 FF 4C B7 3D AD FF 09  73
7749 : C9 0B F0 01 60 AD 00 0A  7C
7751 : 4C 2D 70 A9 00 8D D7 23  97
7759 : 8D ED 23 60 9D EB 23 A9  CB
7761 : 00 A2 07 9D D5 23 9D EB  ED
7769 : 23 60 A9 00 A2 05 9D D5  9B
7771 : 23 9D EB 23 60 A9 00 86  22
7779 : 02 A2 04 9D D5 23 9D EB  46
7781 : 23 60 00 00 00 00 00 00  D5
7789 : 00 00 00 00 00 00 00 00  BA
7791 : 00 00 00 00 00 00 00 00  92
7799 : 00 00 00 00 00 00 00 00  9A
77A1 : 00 00 00 00 00 00 00 00  A2
77A9 : 00 00 00 00 00 00 00 00  AA
77B1 : 00 00 00 00 00 00 00 00  B2
77B9 : 00 00 00 00 00 00 00 00  BA
77C1 : 00 00 00 00 00 00 00 00  C2
77C9 : 00 00 00 00 00 00 00 00  CA
77D1 : 00 00 00 00 00 00 00 00  D2
77D9 : 00 00 00 00 00 00 00 00  DA
77E1 : 00 00 00 00 00 00 00 00  E2
77E9 : 00 00 00 00 00 00 00 00  EA
77F1 : 00 00 00 00 00 00 00 00  F2
77F9 : 00 00 00 00 00 00 00 AE  57
7801 : D3 23 EA EA EA 60 3D 22  69
```

**Listing »Freiheit« (Schluß)**



# Adventure satt!

Für alle Adventure-Freaks, die noch Lust auf mehr Abenteuerspiele haben, geben wir Ihnen hier eine Übersicht über die besten in den 64'er-Sonderheften erschienenen Adventures.

**Zauberschloß:** In diesem Adventure haben Sie die Aufgabe, in einem geheimen Schloß eine sich erst im Spiel herauskristallisierende Aufgabe zu lösen. Natürlich besteht das Schloß aus vielen Gängen, Kammern und einem Kerker. Behindert werden Sie dabei von Ihnen auflauernden Kobolden, die Sie bei falscher Vorgehensweise in einen Zwerg verwandeln. **(SH2/85)**

**Adventure 2000:** Der Erde droht der dritte Weltkrieg. Der Präsident der Vereinigten Staaten beauftragt Sie, den Ernstfall zu verhindern. Sie sollen die Atomrakete, die den Krieg auslösen wird, finden und entschärfen. Ein Chinese kann Ihnen Hilfe geben. Auf Ihrem Weg liegen viele Gefahren und fremde Länder. Nicht zuletzt wird es auch ein Wettlauf gegen die Zeit. **(SH2/85)**

**Quasimodo:** Das Volk der Kartanen hat Sie dazu bestimmt, den tyrannischen Herrscher des Landes aufzuspüren und zur Strecke zu bringen. Leider wissen Sie von ihm nur, daß er Quasimodo heißt und Hexenmeister ist. Als Sie sich bereit erklären, den Auftrag zu übernehmen, werden Sie von Ihren Landsleuten durch einen Geheimgang in sein Gewölbe geführt. **(SH2/85)**

**Mario:** Dieses Adventure verlangt von Ihnen nicht etwa, einen Drachen zu töten oder Schätze zu finden. Sie müssen »nur« ein Höhlensystem durchqueren. Doch es ist alles andere als leicht. Sie werden auf Ihrem Weg von Orks, einer Spinne und dem Ralbog gebremst, wenn nicht getötet. **(SH2/85)**

**Zeittunnel:** Gelangen Sie von Ihrem heimischen Wohnzimmer aus durch den Zeittunnel in die ferne Zukunft. Finden Sie sich auf dem nur von Robotern, Maschinen und Computern bevölkerten Kristallplaneten wieder. Nun möchten Sie wieder nach Hause in Ihr eigenes, bekanntes Wohnzimmer. Doch dazu müssen Sie erst viele Abenteuer in der Zukunft bestehen und noch mehr Rätsel lösen. **(SH2/85)**

**Crantor:** Es geschah einst in ferner Zukunft, als ein garstiger Imperator namens Crantor die Andromeda-Galaxis erobern wollte, indem er sämtliche Imbißstuben zerstören ließ. Das hypergalaktische Imbißdepot beauftragt Sie, den Jäger des verlorenen Hamburgers, den garstigen Herrscher in handliche Portionen zu zerlegen. **(SH2/85)**

**Odyssee:** Dieses Programm ist kein Adventure im üblichen Sinn. Es gehört vielmehr zur Gattung der Fantasy-Rollenspiele wie etwa Ultima II/III oder Questron. Die Reise führt Sie ins Land des Zauberers Saruman, in dem viele Gefahren auf Sie lauern. **(SH2/85)**

**Zauberschloß II:** Der Zauberer, den Sie vor einem Jahr schon einmal vertrieben hatten, ist nach einer Erholungspause zurückgekehrt, um Sie, den neuen König, zu stürzen und sich an Ihnen für seine erlittene Niederlage zu rächen. Nun müssen Sie ihm wieder die Stirn bieten und ihn endgültig besiegen. **(SH3/85)**

**Dark Tower:** Das landschaftlich so schöne Dreistromland droht unter der Herrschaft des Bösen zusammenzubrechen. Helfen Sie der geknechteten Bevölkerung, ihr Dasein wieder in Ruhe und Frieden zu leben. Dabei werden Sie durch Siedlungen und Wälder streifen, Zaubersprüche lernen und zwischen Gut und Böse unterscheiden müssen. **(SH3/85)**

# Impressum

**Herausgeber:** Carl-Franz von Quadt, Otmar Weber

**Chefredakteur:** Michael Scharfenberger
**Leitender Redakteur:** Albert Absmeier
**Koordination:** Georg Klinge
**Redaktion:** Christine Geißler, Boris Schneider, Thomas Röder

**Titelfoto:** Jens Jancke

**Titelgestaltung:** Heinz Rauner Grafik-Design

**Layout:**
Leo Eder (Ltg.), Sigrid Kowalewski (Cheflayouterin), Rolf Raß, Katja Milles

**Herstellung:** Klaus Buck

**Auslandsrepräsentation:**
Schweiz: Markt & Technik Vertriebs AG, Kollerstr. 3, CH-6300 Zug, Tel. 042-41 56 56, Telex: 862 329
USA: M&T Publishing Inc.; 501 Galveston Drive Redwood City, CA 94063 Telefon: (415) 366-3600

**Manuskripteinsendungen:** Manuskripte und Programmlistings werden gerne von der Redaktion angenommen. Sie müssen frei sein von Rechten Dritter. Sollten sie auch an anderer Stelle zur Veröffentlichung oder gewerblichen Nutzung angeboten werden, so muß dies angegeben werden. Mit der Einsendung von Manuskripten und Listings gibt der Verfasser die Zustimmung zum Abdruck in von der Markt & Technik Verlag AG herausgegebenen Publikationen und zur Vervielfältigung der Programmlistings auf Datenträger. Mit der Einsendung von Bauanleitungen gibt der Einsender die Zustimmung zum Abdruck in von Markt & Technik Verlag AG verlegten Publikationen und dazu, daß Markt & Technik Verlag AG Geräte und Bauteile nach der Bauanleitung herstellen läßt und vertreibt oder durch Dritte vertreiben läßt. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen.

**Marketingleiter Vertrieb:** Hans Hörl (114)
**Vertriebsleitung:** Helmut Grünfeldt (189)

**Anzeigenverwaltung und Disposition:** Michaela Hörl

**Verlagsleiter M&T-Buchverlag:** Günther Frank

**Druck:** SOV St.-Otto-Verlag GmbH, Laubanger 23, 8600 Bamberg

**Preis:** Das Einzelheft kostet DM 14,–

**Vertrieb Handelsauflage:** Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz: Pegasus Buch- und Zeitschriften-Vertriebs GmbH, Hauptstätter Straße 96, 7000 Stuttgart 1, Telefon (0711) 648 30



**Verantwortlich:**
Für redaktionellen Teil: Michael Scharfenberger
Für Anzeigen: Britta Fiebig

**Redaktions-Direktor:** Michael M. Pauly

**Vorstand:** Carl-Franz von Quadt, Otmar Weber

**Anschrift für Verlag, Redaktion, Vertrieb, Anzeigenverwaltung und alle Verantwortlichen:**
Markt & Technik Verlag Aktiengesellschaft, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0, Telex 5-22052

# Bücher zum Commodore 64

PROTEXT

TOP

ASS